Image Segmentation Using Deep Learning: A Survey

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos

Abstract—Image segmentation is a key topic in image processing and computer vision with applications such as scene understanding, medical image analysis, robotic perception, video surveillance, augmented reality, and image compression, among many others. Various algorithms for image segmentation have been developed in the literature. Recently, due to the success of deep learning models in a wide range of vision applications, there has been a substantial amount of works aimed at developing image segmentation approaches using deep learning models. In this survey, we provide a comprehensive review of the literature at the time of this writing, covering a broad spectrum of pioneering works for semantic and instance-level segmentation, including fully convolutional pixel-labeling networks, encoder-decoder architectures, multi-scale and pyramid based approaches, recurrent networks, visual attention models, and generative models in adversarial settings. We investigate the similarity, strengths and challenges of these deep learning models, examine the most widely used datasets, report performances, and discuss promising future research directions in this area.

Index Terms—Image segmentation, deep learning, convolutional neural networks, encoder-decoder models, recurrent models, generative models, semantic segmentation, instance segmentation, medical image segmentation.

1 Introduction

Image segmentation is an essential component in many visual understanding systems. It involves partitioning images (or video frames) into multiple segments or objects [1]. Segmentation plays a central role in a broad range of applications [2], including medical image analysis (e.g., tumor boundary extraction and measurement of tissue volumes), autonomous vehicles (e.g., navigable surface and pedestrian detection), video surveillance, and augmented reality to count a few. Numerous image segmentation algorithms have been developed in the literature, from the earliest methods, such as thresholding [3], histogram-based bundling, region-growing [4], k-means clustering [5], watersheds [6], to more advanced algorithms such as active contours [7], graph cuts [8], conditional and Markov random fields [9], and sparsity-based [10]- [11] methods. Over the past few years, however, deep learning (DL) networks have yielded a new generation of image segmentation models with remarkable performance improvements —often achieving the highest accuracy rates on popular benchmarks— resulting in what many regard as a paradigm shift in the field. For example, Figure 1 presents sample image segmentation outputs of a prominent deep learning model, DeepLabv3 [12].

Image segmentation can be formulated as a classification problem of pixels with semantic labels (semantic segmentation) or partitioning of individual objects (instance segmentation). Semantic segmentation performs pixel-level labeling with a set of object categories (e.g., human, car, tree, sky) for all image pixels, thus it is generally a harder undertaking than image classification, which predicts a single label for the entire image. Instance segmentation extends semantic segmentation scope further by detecting and delineating each object of interest in the image (e.g., partitioning of individual persons).

Fig. 1. Segmentation results of DeepLabV3 [12] on sample images.

Our survey covers the most recent literature in image segmentation and discusses more than a hundred deep learning-based segmentation methods proposed until 2019. We provide a comprehensive review and insights on different aspects of these methods, including the training data, the choice of network architectures, loss functions, training strategies, and their key contributions. We present a comparative summary of the performance of the reviewed methods and discuss several challenges and potential future directions for deep learning-based image segmentation models.

We group deep learning-based works into the following categories based on their main technical contributions:

- *S. Minaee is with Snapchat Inc.*
- *Y. Boykov is with the University of Waterloo.*
- *F. Porikli is with the Australian National University, and Huawei.*
- *A. Plaza is with the University of Extremadura, Spain.*
- *N. Kehtarnavaz is with the University of Texas at Dallas.*
- *D. Terzopoulos is with the University of California, Los Angeles.*

1) Fully convolutional networks
2) Convolutional models with graphical models
3) Encoder-decoder based models
4) Multi-scale and pyramid network based models
5) R-CNN based models (for instance segmentation)
6) Dilated convolutional models and DeepLab family
7) Recurrent neural network based models
8) Attention-based models
9) Generative models and adversarial training
10) Convolutional models with active contour models
11) Other models

Some the key contributions of this survey paper can be summarized as follows:

- This survey covers the contemporary literature with respect to segmentation problem, and overviews more than 100 segmentation algorithms proposed till 2019, grouped into 10 categories.
- We provide a comprehensive review and an insightful analysis of different aspects of segmentation algorithms using deep learning, including the training data, the choice of network architectures, loss functions, training strategies, and their key contributions.
- We provide an overview of around 20 popular image segmentation datasets, grouped into 2D, 2.5D (RGB-D), and 3D images.
- We provide a comparative summary of the properties and performance of the reviewed methods for segmentation purposes, on popular benchmarks.
- We provide several challenges and potential future directions for deep learning-based image segmentation.

The remainder of this survey is organized as follows: Section 2 provides an overview of popular deep neural network architectures that serve as the backbone of many modern segmentation algorithms. Section 3 provides a comprehensive overview of the most significant state-of-the-art deep learning based segmentation models, more than 100 till 2019. We also discuss their strengths and contributions over previous works here. Section 4 reviews some of the most popular image segmentation datasets and their characteristics. Section 5.1 reviews popular metrics for evaluating deep-learning-based segmentation models. In Section 5.2, we report the quantitative results and experimental performance of these models. In Section 6, we discuss the main challenges and future directions for deep learning-based segmentation methods. Finally, we present our conclusions in Section 7.

2 Overview of Deep Neural Networks

This section provides an overview of some of the most prominent deep learning architectures used by the computer vision community, including convolutional neural networks (CNNs) [13], recurrent neural networks (RNNs) and long short term memory (LSTM) [14], encoder-decoders [15], and generative adversarial networks (GANs) [16]. With the popularity of deep learning in recent years, several other deep neural architectures have been proposed, such as transformers, capsule networks, gated recurrent units, spatial transformer networks, etc., which will not be covered here.

2.1 Convolutional Neural Networks (CNNs)

CNNs are among the most successful and widely used architectures in the deep learning community, especially for computer vision tasks. CNNs were initially proposed by Fukushima in his seminal paper on the "Neocognitron" [17], based on the hierarchical receptive field model of the visual cortex proposed by Hubel and Wiesel. Subsequently, Waibel *et al.* [18] introduced CNNs with weights shared among temporal receptive fields and backpropagation training for phoneme recognition, and LeCun *et al.* [13] developed a CNN architecture for document recognition (Figure 2).

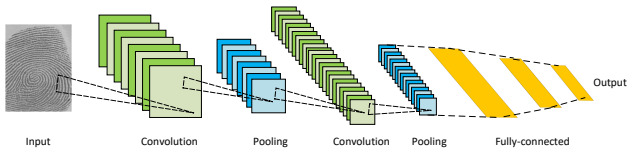


Fig. 2. Architecture of convolutional neural networks. From [13].

CNNs mainly consist of three type of layers: i) convolutional layers, where a kernel (or filter) of weights is convolved in order to extract features; ii) nonlinear layers, which apply an activation function on feature maps (usually element-wise) in order to enable the modeling of non-linear functions by the network; and iii) pooling layers, which replace a small neighborhood of a feature map with some statistical information (mean, max, etc.) about the neighborhood and reduce spatial resolution. The units in layers are locally connected; that is, each unit receives weighted inputs from a small neighborhood, known as the receptive field, of units in the previous layer. By stacking layers to form multi-resolution pyramids, the higher-level layers learn features from increasingly wider receptive fields. The main computational advantage of CNNs is that all the receptive fields in a layer share weights, resulting in a significantly smaller number of parameters than fully-connected neural networks. Some of the most well-known CNN architectures include: AlexNet [19], VGGNet [20], ResNet [21], GoogLeNet [22], MobileNet [23], and DenseNet [24].

2.2 Recurrent Neural Networks (RNNs) and the LSTM

RNNs [25] are widely used to process sequential data, such as speech, text, videos, and time-series, where data at any given time/position depends on previously encountered data. At each time-stamp the model collects the input from the current time X_i and the hidden state from the previous step h_{i-1}, and outputs a target value and a new hidden state (Figure 3).

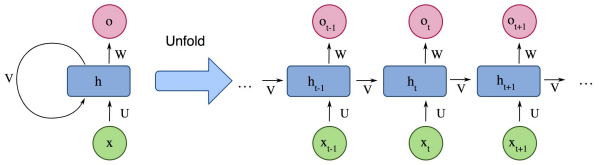


Fig. 3. Architecture of a simple recurrent neural network.

RNNs are typically problematic with long sequences as they cannot capture long-term dependencies in many real-world applications (although they exhibit no theoretical limitations in this regard) and often suffer from gradient vanishing or exploding problems. However, a type of RNNs called Long Short Term Memory (LSTM) [14] is designed to avoid these issues. The LSTM architecture (Figure 4) includes three gates (input gate, output gate, forget gate), which regulate the flow of information into and out from a memory cell, which stores values over arbitrary time intervals.

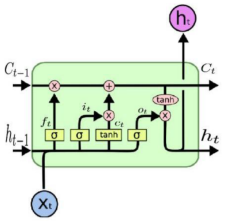


Fig. 4. Architecture of a standard LSTM module. Courtesy of Karpathy.

The relationship between input, hidden states, and different gates is given by:

$$
\begin{aligned}
f_t &= \sigma(\mathbf{W}^{(f)}x_t + \mathbf{U}^{(f)}h_{t-1} + b^{(f)}),\\
i_t &= \sigma(\mathbf{W}^{(i)}x_t + \mathbf{U}^{(i)}h_{t-1} + b^{(i)}),\\
o_t &= \sigma(\mathbf{W}^{(o)}x_t + \mathbf{U}^{(o)}h_{t-1} + b^{(o)}),\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(\mathbf{W}^{(c)}x_t + \mathbf{U}^{(c)}h_{t-1} + b^{(c)}),\\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
\tag{1}
$$

where $x_t \in R^d$ is the input at time-step t, and d denotes the feature dimension for each word, σ denotes the element-wise sigmoid function (to map the values within $[0, 1]$), $\odot$ denotes the element-wise product, and c_t denotes the memory cell designed to lower the risk of vanishing/exploding gradient (and therefore enabling learning of dependencies over larger periods of time, feasible with traditional RNNs). The forget gate, f_t, is intended to reset the memory cell. i_t and o_t denote the input and output gates, respectively, and essentially control the input and output of the memory cell.

2.3 Encoder-Decoder and Auto-Encoder Models

Encoder-Decoder models are a family of models which learn to map data-points from an input domain to an output domain via a two-stage network: The encoder, represented by an encoding function $z = f(x)$, compresses the input into a latent-space representation; the decoder, $y = g(z)$, aims to predict the output from the latent space representation. The latent representation here essentially refers to a feature (vector) representation, which is able to capture the underlying semantic information of the input that is useful for predicting the output. These models are extremely popular in image-to-image translation problems, as well as for sequence models in NLP. Figure 5 illustrates the block-diagram of a simple encoder-decoder model. These models are usually trained by minimizing the reconstruction loss $L(y, \hat{y})$, which measures the differences between the ground-truth output y and the subsequent reconstruction $\hat{y}$. The output here could be an enhanced version of the image (such as in image de-blurring, or super-resolution), or a segmentation map.

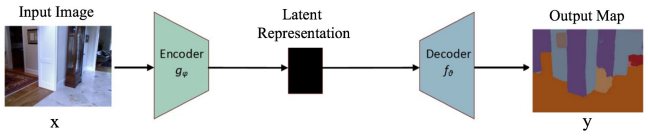


Fig. 5. The architecture of a simple encoder-decoder model.

Auto-encoders are special case of encoder-decoder models in which the input and output are the same. Several variations of auto-encoders have been proposed. One of the most popular is the stacked denoising auto-encoder (SDAE) [26], which stacks several auto-encoders and uses them for image denoising purposes. Another popular variant is the variational auto-encoder (VAE) [27], which imposes a prior distribution on the latent representation. VAEs are able to generate realistic samples from a given data distribution. Another variant is adversarial auto-encoders, which introduces an adversarial loss on the latent representation to encourage them to approximate a prior distribution.

2.4 Generative Adversarial Networks (GANs)

GANs are a newer family of deep learning models [16]. They consist of two networks—a generator and a discriminator (Figure 6). The generator network $G = z \rightarrow y$ in the conventional GAN learns a mapping from noise z (with a prior distribution) to a target distribution y, which is similar to the "real" samples. The discriminator network D attempts to distinguish the generated samples ("fakes") from the "real" ones. The GAN loss function may be written as $\mathcal{L}_{\text{GAN}} = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$. We can regard the GAN as a minimax game between G and D, where D is trying to minimize its classification error in distinguishing fake samples from real ones, hence maximizing the loss function, and G is trying to maximize the discriminator network's error, hence minimizing the loss function. After training the model, the trained generator model would be $G^* = \arg\ \min_G \max_D\ \mathcal{L}_{\text{GAN}}$ In practice, this function may not provide enough gradient for effectively training G, specially initially (when D can easily discriminate fake samples from real ones). Instead of minimizing $\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$, a possible solution is to train it to maximize $\mathbb{E}_{z\sim p_z(z)}[\log(D(G(z)))]$.

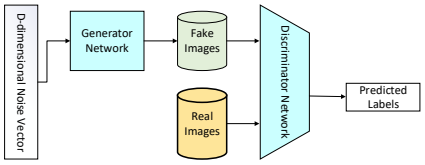


Fig. 6. Architecture of a generative adversarial network.

Since the invention of GANs, researchers have endeavored to improve/modify GANs several ways. For example, Radford *et al.* [28] proposed a convolutional GAN model, which works better than fully-connected networks when used for image generation. Mirza [29] proposed a conditional GAN model that can generate images conditioned on class labels, which enables one to generate samples with specified

labels. Arjovsky *et al.* [30] proposed a new loss function based on the Wasserstein (a.k.a. earth mover's distance) to better estimate the distance for cases in which the distribution of real and generated samples are non-overlapping (hence the KullbackLeiber divergence is not a good measure of the distance). For additional works, we refer the reader to [31].

2.5 Transfer Learning

In some cases the DL-models can be trained from scratch on new applications/datasets (assuming a sufficient quantity of labeled training data), but in many cases there are not enough labeled data available to train a model from scratch and one can use **transfer learning** to tackle this problem. In transfer learning, a model trained on one task is re-purposed on another (related) task, usually by some adaptation process toward the new task. For example, one can imagine adapting an image classification model trained on ImageNet to a different task, such as texture classification, or face recognition. In image segmentation case, many people use a model trained on ImageNet (a larger dataset than most of image segmentation datasets), as the encoder part of the network, and re-train their model from those initial weights. The assumption here is that those pre-trained models should be able to capture the semantic information of the image required for segmentation, and therefore enabling them to train the model with less labeled samples.

3 DL-Based Image Segmentation Models

This section provides a detailed review of more than a hundred deep learning-based segmentation methods proposed until 2019, grouped into 10 categories. It is worth mentioning that there are some pieces that are common among many of these works, such as having encoder and decoder parts, skip-connections, multi-scale analysis, and more recently the use of dilated convolution. Because of this, it is difficult to mention the unique contributions of each work, but easier to group them based on their underlying architectural contribution over previous works.

3.1 Fully Convolutional Networks

Long *et al.* [32] proposed one of the first deep learning works for semantic image segmentation, using a fully convolutional network (FCN). An FCN (Figure 7) includes only convolutional layers, which enables it to take an image of arbitrary size and produce a segmentation map of the same size. The authors modified existing CNN architectures, such as VGG16 and GoogLeNet, to manage non-fixed sized input and output, by replacing all fully-connected layers with the fully-convolutional layers. As a result, the model outputs a spatial segmentation map instead of classification scores.

Through the use of skip connections in which feature maps from the final layers of the model are up-sampled and fused with feature maps of earlier layers (Figure 8), the model combines semantic information (from deep, coarse layers) and appearance information (from shallow, fine layers) in order to produce accurate and detailed segmentations. The model was tested on PASCAL VOC, NYUDv2, and SIFT Flow, and achieved state-of-the-art segmentation performance.

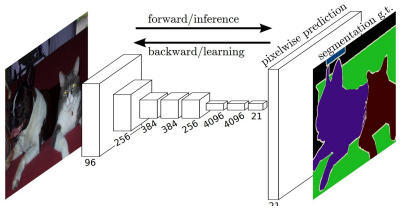


Fig. 7. A fully convolutional image segmentation network. The FCN learns to make dense, pixel-wise predictions. From [32].

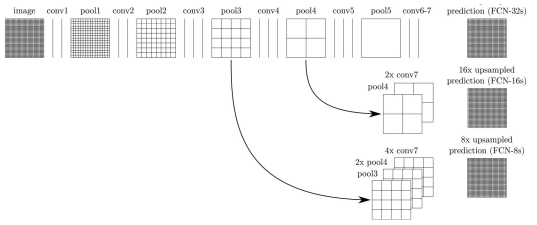


Fig. 8. Skip connections combine coarse, high-level information and fine, low-level information. From [32].

This work is considered a milestone in image segmentation, demonstrating that deep networks can be trained for semantic segmentation in an end-to-end manner on variable-sized images. However, despite its popularity and effectiveness, the conventional FCN model has some limitations—it is not fast enough for real-time inference, it does not take into account the global context information in an efficient way, and it is not easily transferable to 3D images. Several efforts have attempted to overcome some of the limitations of the FCN.

For instance, Liu *et al.* [33] proposed a model called ParseNet, to address an issue with FCN—ignoring global context information. ParseNet adds global context to FCNs by using the average feature for a layer to augment the features at each location. The feature map for a layer is pooled over the whole image resulting in a context vector. This context vector is normalized and unpooled to produce new feature maps of the same size as the initial ones. These feature maps are then concatenated. In a nutshell, ParseNet is an FCN with the described module replacing the convolutional layers (Figure 9).

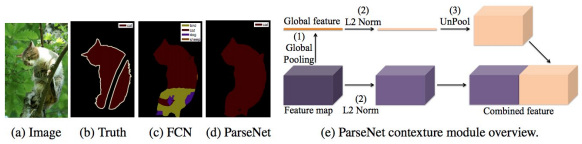


Fig. 9. ParseNet, showing the use of extra global context to produce smoother segmentation (d) than an FCN (c). From [33].

FCNs have been applied to a variety of segmentation problems, such as brain tumor segmentation [34], instance-aware semantic segmentation [35], skin lesion segmentation [36], and iris segmentation [37].

3.2 Convolutional Models With Graphical Models

As discussed, FCN ignores potentially useful scene-level semantic context. To integrate more context, several ap-

proaches incorporate probabilistic graphical models, such as Conditional Random Fields (CRFs) and Markov Random Field (MRFs), into DL architectures.

Chen *et al.* [38] proposed a semantic segmentation algorithm based on the combination of CNNs and fully connected CRFs (Figure 10). They showed that responses from the final layer of deep CNNs are not sufficiently localized for accurate object segmentation (due to the invariance properties that make CNNs good for high level tasks such as classification). To overcome the poor localization property of deep CNNs, they combined the responses at the final CNN layer with a fully-connected CRF. They showed that their model is able to localize segment boundaries at a higher accuracy rate than it was possible with previous methods.

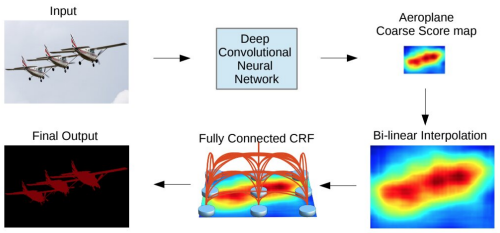


Fig. 10. A CNN+CRF model. The coarse score map of a CNN is upsampled via interpolated interpolation, and fed to a fully-connected CRF to refine the segmentation result. From [38].

Schwing and Urtasun [39] proposed a fully-connected deep structured network for image segmentation. They presented a method that jointly trains CNNs and fully-connected CRFs for semantic image segmentation, and achieved encouraging results on the challenging PASCAL VOC 2012 dataset. In [40], Zheng et al. proposed a similar semantic segmentation approach integrating CRF with CNN.

In another relevant work, Lin *et al.* [41] proposed an efficient algorithm for semantic segmentation based on contextual deep CRFs. They explored "patch-patch" context (between image regions) and "patch-background" context to improve semantic segmentation through the use of contextual information.

Liu *et al.* [42] proposed a semantic segmentation algorithm that incorporates rich information into MRFs, including high-order relations and mixture of label contexts. Unlike previous works that optimized MRFs using iterative algorithms, they proposed a CNN model, namely a Parsing Network, which enables deterministic end-to-end computation in a single forward pass.

3.3 Encoder-Decoder Based Models

Another popular family of deep models for image segmentation is based on the convolutional encoder-decoder architecture. Most of the DL-based segmentation works use some kind of encoder-decoder models. We group these works into two categories, encoder-decoder models for general segmentation, and for medical image segmentation (to better distinguish between applications).

3.3.1 Encoder-Decoder Models for General Segmentation

Noh *et al.* [43] published an early paper on semantic segmentation based on deconvolution (a.k.a. transposed convolution). Their model (Figure 11) consists of two parts, an encoder using convolutional layers adopted from the VGG 16-layer network and a deconvolutional network that takes the feature vector as input and generates a map of pixel-wise class probabilities. The deconvolution network is composed of deconvolution and unpooling layers, which identify pixel-wise class labels and predict segmentation masks. This network achieved promising performance on the PASCAL VOC 2012 dataset, and obtained the best accuracy (72.5%) among the methods trained with no external data at the time.

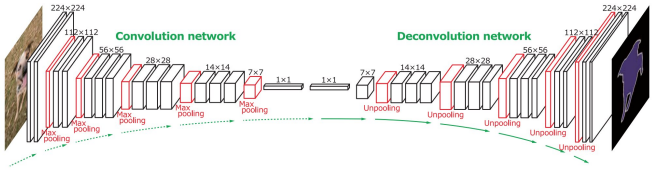


Fig. 11. Deconvolutional semantic segmentation. Following a convolution network based on the VGG 16-layer net, is a multi-layer deconvolution network to generate the accurate segmentation map. From [43].

In another promising work known as SegNet, Badrinarayanan *et al.* [44] proposed a convolutional encoder-decoder architecture for image segmentation (Figure 12). Similar to the deconvolution network, the core trainable segmentation engine of SegNet consists of an encoder network, which is topologically identical to the 13 convolutional layers in the VGG16 network, and a corresponding decoder network followed by a pixel-wise classification layer. The main novelty of SegNet is in the way the decoder upsamples its lower resolution input feature map(s); specifically, it uses pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to up-sample. The (sparse) up-sampled maps are then convolved with trainable filters to produce dense feature maps. SegNet is also significantly smaller in the number of trainable parameters than other competing architectures. A Bayesian version of SegNet was also proposed by the same authors to model the uncertainty inherent to the convolutional encoder-decoder network for scene segmentation [45].

Another popular model in this category is the recently-developed segmentation network, high-resolution network (HRNet) [119] Figure 13. Other than recovering high-resolution representations as done in DeConvNet, SegNet, U-Net and V-Net, HRNet maintains high-resolution representations through the encoding process by connecting the high-to-low resolution convolution streams in parallel, and repeatedly exchanging the information across resolutions.

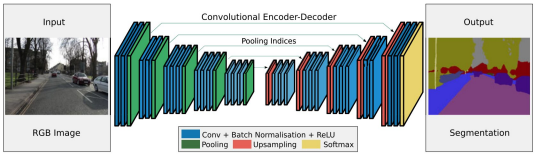


Fig. 12. SegNet has no fully-connected layers; hence, the model is fully convolutional. A decoder up-samples its input using the transferred pool indices from its encoder to produce a sparse feature map(s). From [44].

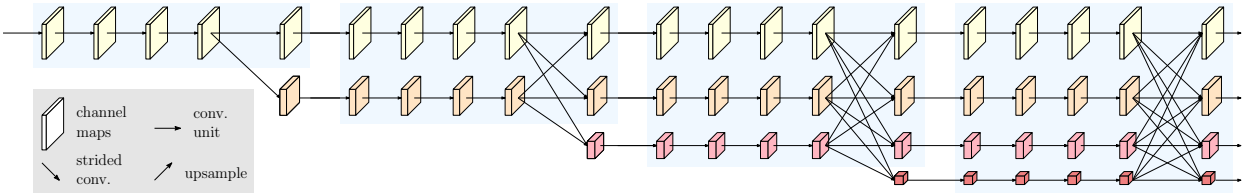


Fig. 13. Illustrating the HRNet architecture. It consists of parallel high-to-low resolution convolution streams with repeated information exchange across multi-resolution steams. There are four stages. The 1st stage consists of high-resolution convolutions. The 2nd (3rd, 4th) stage repeats two-resolution (three-resolution, four-resolution) blocks. From [119].

Many of the more recent works on semantic segmentation use HRNet as the backbone by exploiting contextual models, such as self-attention and its extensions.

Several other works adopt transposed convolutions, or encoder-decoders for image segmentation, such as Stacked Deconvolutional Network (SDN) [46], Linknet [47], W-Net [48], and locality-sensitive deconvolution networks for RGB-D segmentation [49].

3.3.2 Encoder-Decoder Models for Medical and Biomedical Image Segmentation

There are several models initially developed for medical/biomedical image segmentation, which are inspired by FCNs and encoder-decoder models. U-Net [50], and V-Net [51], are two well-known such architectures, which are now also being used outside the medical domain.

Ronneberger *et al.* [50] proposed the U-Net for segmenting biological microscopy images. Their network and training strategy relies on the use of data augmentation to learn from the available annotated images more effectively. The U-Net architecture (Figure 14) comprises two parts, a contracting path to capture context, and a symmetric expanding path that enables precise localization. The down-sampling or contracting part has a FCN-like architecture that extracts features with 3×3 convolutions. The up-sampling or expanding part uses up-convolution (or deconvolution), reducing the number of feature maps while increasing their dimensions. Feature maps from the down-sampling part of the network are copied to the up-sampling part to avoid losing pattern information. Finally, a 1×1 convolution processes the feature maps to generate a segmentation map that categorizes each pixel of the input image. U-Net was trained on 30 transmitted light microscopy images, and it won the ISBI cell tracking challenge 2015 by a large margin.

Various extensions of U-Net have been developed for different kinds of images. For example, Cicek [52] proposed a U-Net architecture for 3D images. Zhou *et al.* [53] developed a nested U-Net architecture. U-Net has also been applied to various other problems. For example, Zhang *et al.* [54] developed a road segmentation/extraction algorithm based on U-Net.

V-Net (Figure 15) is another well-known, FCN-based model, which was proposed by Milletari *et al.* [51] for 3D medical image segmentation. For model training, they introduced a new objective function based on the Dice coefficient, enabling the model to deal with situations in which there is a strong imbalance between the number of voxels in the foreground and background. The network was trained end-to-end on MRI volumes depicting prostate, and learns to predict segmentation for the whole volume at once.

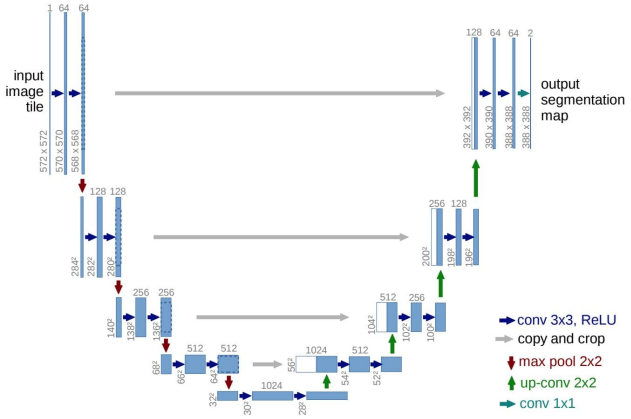


Fig. 14. The U-net model. The blue boxes denote feature map blocks with their indicated shapes. From [50].

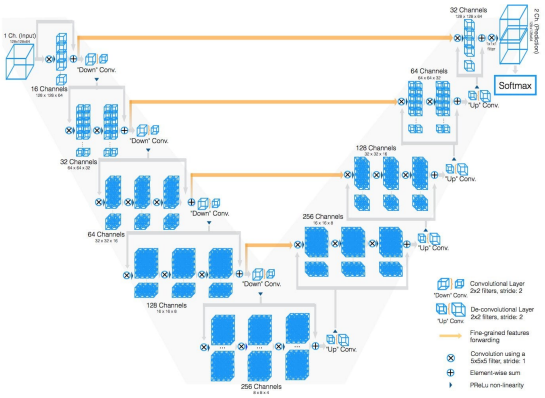


Fig. 15. The V-net model for 3D image segmentation. From [51].

Some of the other relevant works on medical image segmentation includes Progressive Dense V-net (PDV-Net) *et al.* for fast and automatic segmentation of pulmonary lobes from chest CT images, and the 3D-CNN encoder for lesion segmentation [55].

3.4 Multi-Scale and Pyramid Network Based Models

Multi-scale analysis, a rather old idea in image processing, has been deployed in various neural network architectures. One of the most prominent models of this sort is the Feature Pyramid Network (FPN) proposed by Lin *et al.* [56], which was developed mainly for object detection but was then also

applied to segmentation. The inherent multi-scale, pyramidal hierarchy of deep CNNs was used to construct feature pyramids with marginal extra cost. To merge low and high resolution features, the FPN is composed of a bottom-up pathway, a top-down pathway and lateral connections. The concatenated feature maps are then processed by a 3×3 convolution to produce the output of each stage. Finally, each stage of the top-down pathway generates a prediction to detect an object. For image segmentation, the authors use two multi-layer perceptrons (MLPs) to generate the masks. Figure 16 shows how the lateral connections and the top-down pathway are merged via addition.

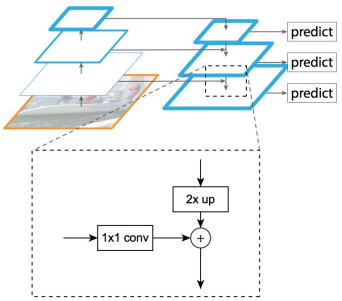


Fig. 16. A building block illustrating the lateral connection and the top-down pathway, merged by addition. From [56].

Zhao *et al.* [57] developed the Pyramid Scene Parsing Network (PSPN), a multi-scale network to better learn the global context representation of a scene (Figure 17). Different patterns are extracted from the input image using a residual network (ResNet) as a feature extractor, with a dilated network. These feature maps are then fed into a pyramid pooling module to distinguish patterns of different scales. They are pooled at four different scales, each one corresponding to a pyramid level and processed by a 1×1 convolutional layer to reduce their dimensions. The outputs of the pyramid levels are up-sampled and concatenated with the initial feature maps to capture both local and global context information. Finally, a convolutional layer is used to generate the pixel-wise predictions.

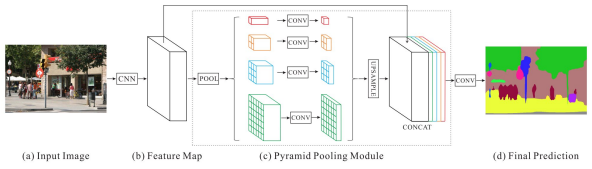


Fig. 17. The PSPN architecture. A CNN produces the feature map and a pyramid pooling module aggregates the different sub-region representations. Up-sampling and concatenation are used to form the final feature representation from which, the final pixel-wise prediction is obtained through convolution. From [57].

Ghiasi and Fowlkes [58] developed a multi-resolution reconstruction architecture based on a Laplacian pyramid that uses skip connections from higher resolution feature maps and multiplicative gating to successively refine segment boundaries reconstructed from lower-resolution maps. They showed that, while the apparent spatial resolution of convolutional feature maps is low, the high-dimensional feature representation contains significant sub-pixel localization information.

There are other models using multi-scale analysis for segmentation, such as DM-Net (Dynamic Multi-scale Filters Network) [59], Context contrasted network and gated multi-scale aggregation (CCN) [60], Adaptive Pyramid Context Network (APC-Net) [61], Multi-scale context intertwining (MSCI) [62], and salient object segmentation [63].

3.5 R-CNN Based Models (for Instance Segmentation)

The regional convolutional network (R-CNN) and its extensions (Fast R-CNN, Faster R-CNN, Mask-RCNN) have proven successful in object detection applications. Some of the extensions of R-CNN have been heavily used to address the instance segmentation problem; i.e., the task of simultaneously performing object detection and semantic segmentation. In particular, the Faster R-CNN [64] architecture (Figure 18) developed for object detection uses a region proposal network (RPN) to propose bounding box candidates. The RPN extracts a Region of Interest (RoI), and a RoIPool layer computes features from these proposals in order to infer the bounding box coordinates and the class of the object.

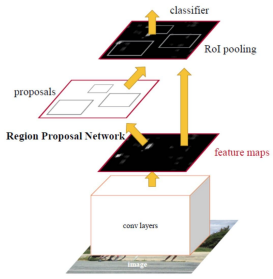


Fig. 18. Faster R-CNN architecture. Each image is processed by convolutional layers and its features are extracted, a sliding window is used in RPN for each location over the feature map, for each location, k ($k = 9$) anchor boxes are used (3 scales of 128, 256 and 512, and 3 aspect ratios of 1:1, 1:2, 2:1) to generate a region proposal; A cls layer outputs 2k scores whether there or not there is an object for k boxes; A reg layer outputs $4k$ for the coordinates (box center coordinates, width and height) of k boxes. From [64].

In one extension of this model, He *et al.* [65] proposed a Mask R-CNN for object instance segmentation, which beat all previous benchmarks on many COCO challenges. This model efficiently detects objects in an image while simultaneously generating a high-quality segmentation mask for each instance. Mask R-CNN is essentially a Faster R-CNN with 3 output branches (Figure 19)—the first computes the bounding box coordinates, the second computes the associated classes, and the third computes the binary mask to segment the object. The Mask R-CNN loss function combines the losses of the bounding box coordinates, the predicted class, and the segmentation mask, and trains all of them jointly. Figure 20 shows the Mask-RCNN result on some sample images.

The Path Aggregation Network (PANet) proposed by Liu *et al.* [66] is based on the Mask R-CNN and FPN models (Figure 21). The feature extractor of the network uses an FPN architecture with a new augmented bottom-up pathway improving the propagation of low-layer features. Each stage

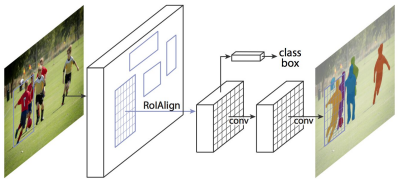


Fig. 19. Mask R-CNN architecture for instance segmentation. From [65].

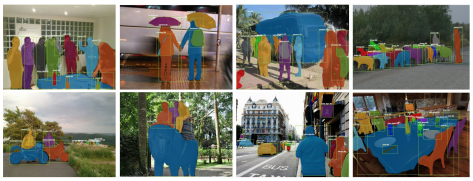


Fig. 20. Mask R-CNN results on sample images from the COCO test set. From [65].

of this third pathway takes as input the feature maps of the previous stage and processes them with a 3×3 convolutional layer. The output is added to the same stage feature maps of the top-down pathway using a lateral connection and these feature maps feed the next stage. As in the Mask R-CNN, the output of the adaptive feature pooling layer feeds three branches. The first two use a fully connected layer to generate the predictions of the bounding box coordinates and the associated object class. The third processes the RoI with an FCN to predict the object mask.

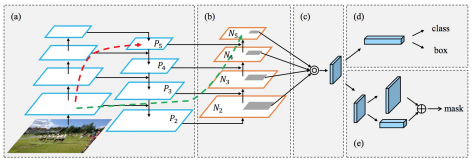


Fig. 21. The Path Aggregation Network. (a) FPN backbone. (b) Bottom-up path augmentation. (c) Adaptive feature pooling. (d) Box branch. (e) Fully-connected fusion. Courtesy of [66].

Dai *et al.* [67] developed a multi-task network for instance-aware semantic segmentation, that consists of three networks, respectively differentiating instances, estimating masks, and categorizing objects. These networks form a cascaded structure, and are designed to share their convolutional features. Hu *et al.* [68] proposed a new partially-supervised training paradigm, together with a novel weight transfer function, that enables training instance segmentation models on a large set of categories, all of which have box annotations, but only a small fraction of which have mask annotations.

Chen *et al.* [69] developed an instance segmentation model, MaskLab (Figure 22), by refining object detection with semantic and direction features based on Faster R-CNN. This model produces three outputs, box detection, semantic segmentation, and direction prediction. Building on the Faster-RCNN object detector, the predicted boxes provide accurate localization of object instances. Within each region of interest, MaskLab performs foreground/background segmentation by combining semantic and direction prediction.

Another interesting model is Tensormask, proposed by Chen *et al.* [70], which is based on dense sliding window

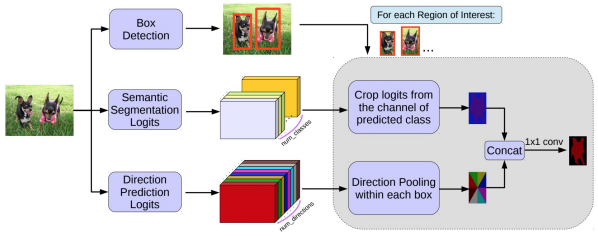


Fig. 22. The MaskLab model. MaskLab generates three outputs—refined box predictions (from Faster R-CNN), semantic segmentation logits for pixel-wise classification, and direction prediction logits for predicting each pixel's direction toward its instance center. From [69].

instance segmentation. They treat dense instance segmentation as a prediction task over 4D tensors and present a general framework that enables novel operators on 4D tensors. They demonstrate that the tensor view leads to large gains over baselines and yields results comparable to Mask R-CNN. TensorMask achieves promising results on dense object segmentation (Figure 23).

Fig. 23. The predicted segmentation map of a sample image by TensorMask. From [70].

Many other instance segmentation models have been developed based on R-CNN, such as those developed for mask proposals, including R-FCN [71], DeepMask [72], SharpMask [73], PolarMask [74], and boundary-aware instance segmentation [75]. It is worth noting that there is another promising research direction that attempts to solve the instance segmentation problem by learning grouping cues for bottom-up segmentation, such as Deep Watershed Transform [76], and Semantic Instance Segmentation via Deep Metric Learning [77].

3.6 Dilated Convolutional Models and DeepLab Family

Dilated convolution (a.k.a. "atrous" convolution) introduces another parameter to convolutional layers, the dilation rate. The dilated convolution (Figure 24) of a signal $x(i)$ is defined as $y_i = \sum_{k=1}^{K} x[i + rk]w[k]$, where r is the dilation rate that defines a spacing between the weights of the kernel w. For example, a 3×3 kernel with a dilation rate of 2 will have the same size receptive field as a 5×5 kernel while using only 9 parameters, thus enlarging the receptive field with no increase in computational cost. Dilated convolutions have been popular in the field of real-time segmentation, and many recent publications report the use of this technique. Some of most important include the DeepLab family [78], multi-scale context aggregation [79], dense upsampling convolution and hybrid dilatedconvolution (DUC-HDC) [80], densely

connected Atrous Spatial Pyramid Pooling (DenseASPP) [81], and the efficient neural network (ENet) [82].

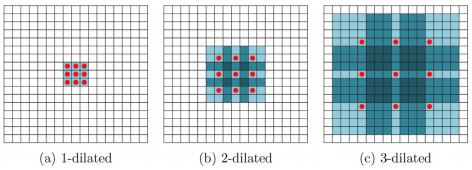


Fig. 24. Dilated convolution. A 3×3 kernel at different dilation rates.

DeepLabv1 [38] and DeepLabv2 [78] are among some of the most popular image segmentation approaches, developed by Chen *et al.*. The latter has three key features. First is the use of dilated convolution to address the decreasing resolution in the network (caused by max-pooling and striding). Second is Atrous Spatial Pyramid Pooling (ASPP), which probes an incoming convolutional feature layer with filters at multiple sampling rates, thus capturing objects as well as image context at multiple scales to robustly segment objects at multiple scales. Third is improved localization of object boundaries by combining methods from deep CNNs and probabilistic graphical models. The best DeepLab (using a ResNet-101 as backbone) has reached a 79.7% mIoU score on the 2012 PASCAL VOC challenge, a 45.7% mIoU score on the PASCAL-Context challenge and a 70.4% mIoU score on the Cityscapes challenge. Figure 25 illustrates the Deeplab model, which is similar to [38], the main difference being the use of dilated convolution and ASPP.

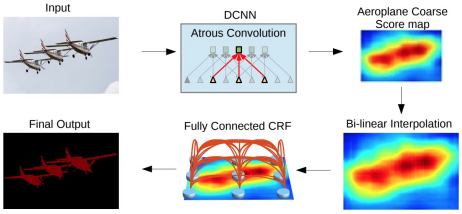


Fig. 25. The DeepLab model. A CNN model such as VGG-16 or ResNet-101 is employed in fully convolutional fashion, using dilated convolution. A bilinear interpolation stage enlarges the feature maps to the original image resolution. Finally, a fully connected CRF refines the segmentation result to better capture the object boundaries. From [78]

Subsequently, Chen *et al.* [12] proposed DeepLabv3, which combines cascaded and parallel modules of dilated convolutions. The parallel convolution modules are grouped in the ASPP. A 1×1 convolution and batch normalisation are added in the ASPP. All the outputs are concatenated and processed by another 1×1 convolution to create the final output with logits for each pixel.

In 2018, Chen *et al.* [83] released Deeplabv3+, which uses an encoder-decoder architecture (Figure 26), including atrous 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). They used the DeepLabv3 framework as encoder. The most relevant model has a modified Xception backbone with more layers, dilated depthwise separable convolutions instead of max pooling and batch normalization. The best DeepLabv3+ pretrained on the COCO and the JFT datasets has obtained a 89.0% mIoU score on the 2012 PASCAL VOC challenge.

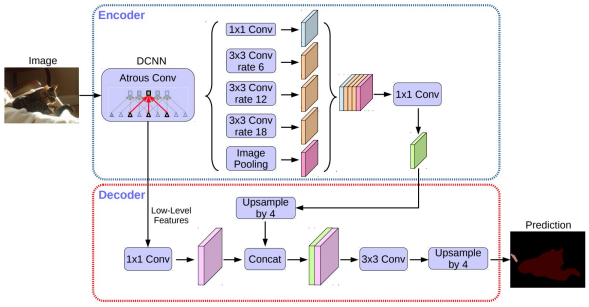


Fig. 26. The DeepLabv3+ model. From [83].

3.7 Recurrent Neural Network Based Models

While CNNs are a natural fit for computer vision problems, they are not the only possibility. RNNs are useful in modeling the short/long term dependencies among pixels to (potentially) improve the estimation of the segmentation map. Using RNNs, pixels may be linked together and processed sequentially to model global contexts and improve semantic segmentation. One challenge, though, is the natural 2D structure of images.

Visin *et al.* [84] proposed an RNN-based model for semantic segmentation called ReSeg. This model is mainly based on another work, ReNet [85], which was developed for image classification. Each ReNet layer (Figure 27) is composed of four RNNs that sweep the image horizontally and vertically in both directions, encoding patches/activations, and providing relevant global information. To perform image segmentation with the ReSeg model (Figure 28), ReNet layers are stacked on top of pre-trained VGG-16 convolutional layers that extract generic local features. ReNet layers are then followed by up-sampling layers to recover the original image resolution in the final predictions. Gated Recurrent Units (GRUs) are used because they provide a good balance between memory usage and computational power.

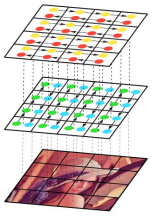


Fig. 27. A single-layer ReNet. From [85].

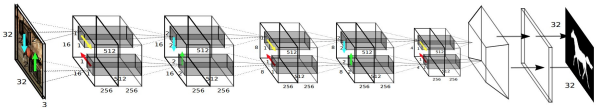


Fig. 28. The ReSeg model. The pre-trained VGG-16 feature extractor network is not shown. From [84].

In another work, Byeon *et al.* [86] developed a pixel-level segmentation and classification of scene images using long-short-term-memory (LSTM) network. They investigated two-dimensional (2D) LSTM networks for images of natural scenes, taking into account the complex spatial dependencies of labels. In this work, classification, segmentation, and context integration are all carried out by 2D LSTM networks, allowing texture and spatial model parameters to be learned within a single model. The block-diagram of the proposed 2D LSTM network for image segmentation in [86] is shown in Figure 29.

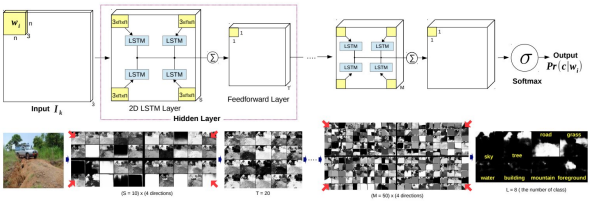


Fig. 29. The 2D-LSTM model for semantic segmentation. The input image is divided into non-overlapping windows. Each window with RGB channels ($3 \times N \times N$) is fed into four separate LSTM memory blocks. The current window of LSTM block is connected to its surrounding directions x and y; i.e., left-top, left-bottom, right-top, and right-bottom; it propagates surrounding contexts. The output of each LSTM block is then passed to the feedforward layer, that sums all directions and applies hyperbolic tangent. In the final layer, the outputs of the final LSTM blocks are summed up and sent to the softmax layer. From [86].

Liang *et al.* [87] proposed a semantic segmentation model based on the Graph Long Short-Term Memory (Graph LSTM) network, a generalization of LSTM from sequential data or multidimensional data to general graph-structured data. Instead of evenly dividing an image to pixels or patches in existing multi-dimensional LSTM structures (e.g., row, grid and diagonal LSTMs), they take each arbitrary-shaped superpixel as a semantically consistent node, and adaptively construct an undirected graph for the image, where the spatial relations of the superpixels are naturally used as edges. Figure 30 presents a visual comparison of the traditional pixel-wise RNN model and graph-LSTM model. To adapt the Graph LSTM model to semantic segmentation (Figure 31), LSTM layers built on a super-pixel map are appended on the convolutional layers to enhance visual features with global structure context. The convolutional features pass through 1×1 convolutional filters to generate the initial confidence maps for all labels. The node updating sequence for the subsequent Graph LSTM layers is determined by the confidence-drive scheme based on the initial confidence maps, and then the Graph LSTM layers can sequentially update the hidden states of all superpixel nodes.

Xiang and Fox [88] proposed Data Associated Recurrent Neural Networks (DA-RNNs), for joint 3D scene mapping and semantic labeling. DA-RNNs use a new recurrent neural network architecture (Figure 32) for semantic labeling on RGB-D videos. The output of the network is integrated with mapping techniques such as Kinect-Fusion in order to inject semantic information into the reconstructed 3D scene.

Hu *et al.* [89] developed a semantic segmentation algorithm based on natural language expression, using a combination of CNN to encode the image and LSTM to encode its natural language description. This is different

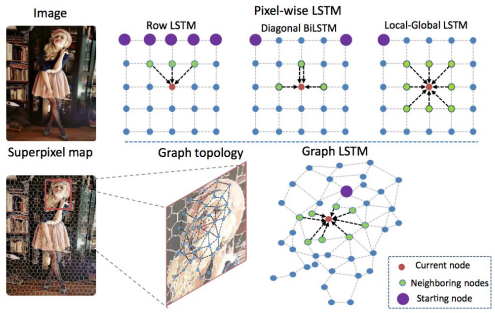


Fig. 30. Comparison between the graph-LSTM model and traditional pixel-wise RNN models. From [87].

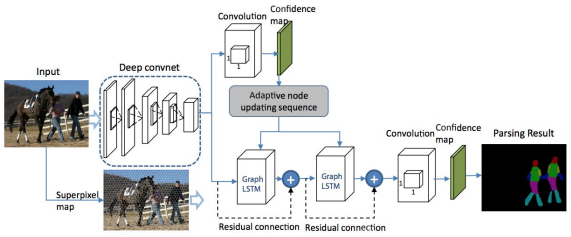


Fig. 31. The graph-LSTM model for semantic segmentation. From [87].

from traditional semantic segmentation over a predefined set of semantic classes, as, e.g., the phrase "two men sitting on the right bench" requires segmenting only the two people on the right bench and no one standing or sitting on another bench. To produce pixel-wise segmentation for language expression, they propose an end-to-end trainable recurrent and convolutional model that jointly learns to process visual and linguistic information (Figure 33). In the considered model, a recurrent LSTM network is used to encode the referential expression into a vector representation, and an FCN is used to extract a spatial feature map from the image and output a spatial response map for the target object. An example segmentation result of this model (for the query "people in blue coat") is shown in Figure 34.

3.8 Attention-Based Models

Attention mechanisms have been persistently explored in computer vision over the years, and it is therefore not surprising to find publications that apply such mechanisms to semantic segmentation.

Chen *et al.* [90] proposed an attention mechanism that learns to softly weight multi-scale features at each pixel

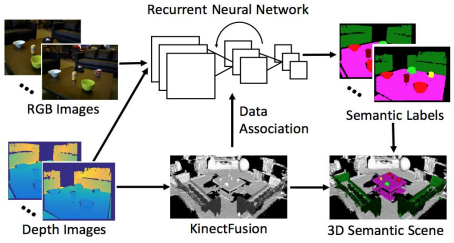


Fig. 32. The DA-RNN architecture. From [88].

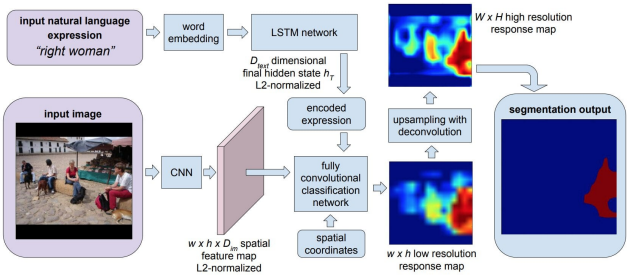


Fig. 33. The CNN+LSTM architecture for segmentation from natural language expressions. From [89].

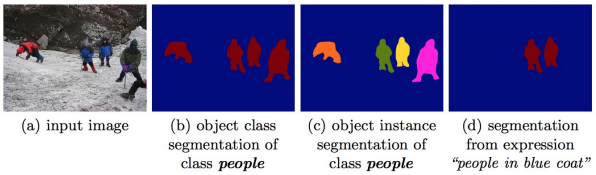


Fig. 34. Segmentation masks generated for the query "people in blue coat". From [89].

location. They adapt a powerful semantic segmentation model and jointly train it with multi-scale images and the attention model (Figure 35). The attention mechanism outperforms average and max pooling, and it enables the model to assess the importance of features at different positions and scales.

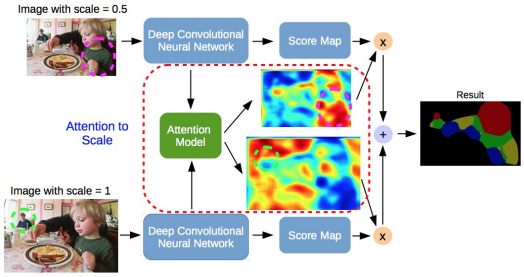


Fig. 35. Attention-based semantic segmentation model. The attention model learns to assign different weights to objects of different scales; e.g., the model assigns large weights on the small person (green dashed circle) for features from scale 1.0, and large weights on the large child (magenta dashed circle) for features from scale 0.5. From [90].

In contrast to other works in which convolutional classifiers are trained to learn the representative semantic features of labeled objects, Huang *et al.* [91] proposed a semantic segmentation approach using reverse attention mechanisms. Their Reverse Attention Network (RAN) architecture (Figure 36) trains the model to capture the opposite concept (i.e., features that are not associated with a target class) as well. The RAN is a three-branch network that performs the direct, and reverse-attention learning processes simultaneously.

Li *et al.* [92] developed a Pyramid Attention Network for semantic segmentation. This model exploits the impact of global contextual information in semantic segmentation. They combined attention mechanisms and spatial pyramids to extract precise dense features for pixel labeling, instead of complicated dilated convolutions and artificially designed

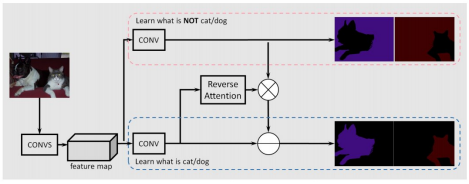


Fig. 36. The reverse attention network for segmentation. From [91].

decoder networks.

More recently, Fu *et al.* [93] proposed a dual attention network for scene segmentation, which can capture rich contextual dependencies based on the self-attention mechanism. Specifically, they append two types of attention modules on top of a dilated FCN which models the semantic interdependencies in spatial and channel dimensions, respectively. The position attention module selectively aggregates the feature at each position by a weighted sum of the features at all positions. The architecture of the dual attention network is shown in Figure 37.

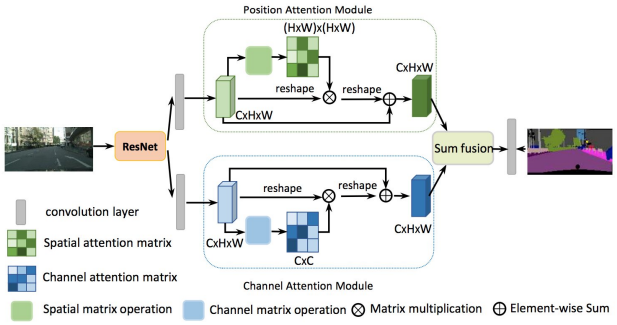


Fig. 37. The dual attention network for semantic segmentation. Courtesy of [93].

Various other works explore attention mechanisms for semantic segmentation, such as OCNet [94] which proposed an object context pooling inspired by self-attention mechanism, Expectation-Maximization Attention (EMANet) [95], Criss-Cross Attention Network (CCNet) [96], end-to-end instance segmentation with recurrent attention [97], a point-wise spatial attention network for scene parsing [98], and a discriminative feature network (DFN) [99], which comprises two sub-networks: a Smooth Network (that contains a Channel Attention Block and global average pooling to select the more discriminative features) and a Border Network (to make the bilateral features of the boundary distinguishable).

3.9 Generative Models and Adversarial Training

Since their introduction, GANs have been applied to a wide range tasks in computer vision, and have been adopted for image segmentation too.

Luc *et al.* [100] proposed an adversarial training approach for semantic segmentation. They trained a convolutional semantic segmentation network (Figure 38), along with an adversarial network that discriminates ground-truth segmentation maps from those generated by the segmentation network. They showed that the adversarial training approach leads to improved accuracy on the Stanford Background and PASCAL VOC 2012 datasets.

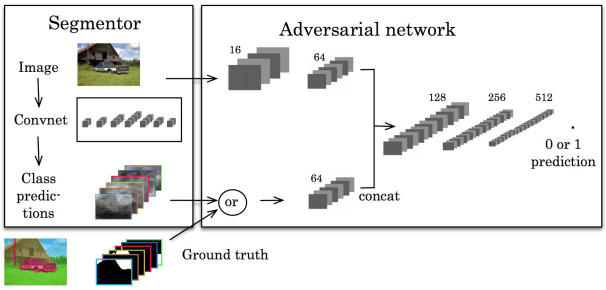


Fig. 38. The proposed adversarial model for semantic segmentation. The segmentation network (left) inputs an RGB image and produces per-pixel class predictions. The adversarial network (right) inputs the label map and produces class labels (1=ground truth or 0=synthetic). From [100].

Figure 39 shows the improvement brought up by adversarial training on one example image from Stanford Background dataset.

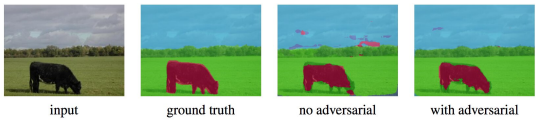


Fig. 39. Segmentation result on a sample image from Stanford Background with and without adversarial training. From [100].

Souly *et al.* [101] proposed semi-weakly supervised semantic segmentation using GANs. It consists of a generator network providing extra training examples to a multi-class classifier, acting as discriminator in the GAN framework, that assigns sample a label y from the K possible classes or marks it as a fake sample (extra class).

In another work, Hung *et al.* [102] developed a framework for semi-supervised semantic segmentation using an adversarial network. They designed an FCN discriminator to differentiate the predicted probability maps from the ground truth segmentation distribution, considering the spatial resolution. The considered loss function of this model contains three terms: cross-entropy loss on the segmentation ground truth, adversarial loss of the discriminator network, and semi-supervised loss based on the confidence map; i.e., the output of the discriminator. The architecture of the model by Hung and colleagues is shown in Figure 40.

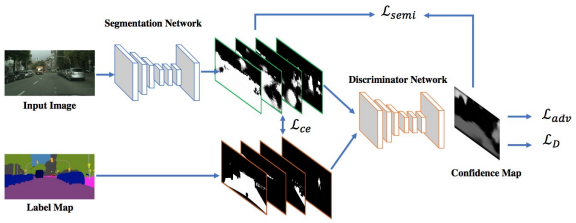


Fig. 40. A semi-supervised segmentation framework. From [102].

Xue *et al.* [103] proposed an adversarial network with multi-scale L1 Loss for medical image segmentation. They used an FCN as the segmentor to generate segmentation label maps, and proposed a novel adversarial critic network with a multi-scale L1 loss function to force the critic and segmentor to learn both global and local features that capture long and short range spatial relationships between pixels. The block-diagram of the segmentor and critic networks are shown in Figure 41.

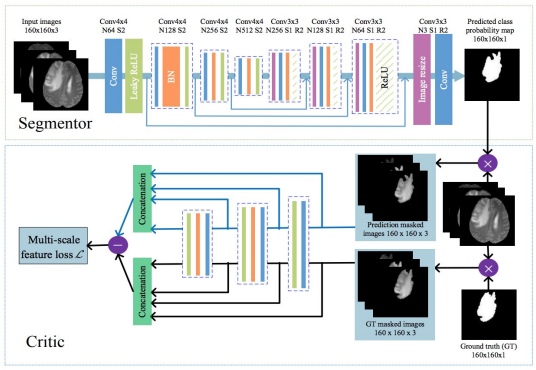


Fig. 41. The proposed adversarial network with multi-scale L1 Loss for semantic segmentation. From [103].

Various other publications report on segmentation models based on adversarial training, such as Cell Image Segmentation Using GANs [104], and segmentation and generation of the invisible parts of objects [105].

3.10 CNN Models With Active Contour Models

The exploration of synergies between FCNs and Active Contour Models (ACMs) [7] has recently attracted research interest. One approach is to formulate new loss functions that are inspired by ACM principles. For example, inspired by the global energy formulation of [106], Chen *et al.* [107] proposed a supervised loss layer that incorporated area and size information of the predicted masks during training of an FCN and tackled the problem of ventricle segmentation in cardiac MRI. Similarly, Gur *et al.* [108] presented an unsupervised loss function based on morphological active contours without edges [109] for microvascular image segmentation.

A different approach initially sought to utilize the ACM merely as a post-processor of the output of an FCN and several efforts attempted modest co-learning by pre-training the FCN. One example of an ACM post-processor for the task of semantic segmentation of natural images is the work by Le *et al.* [110] in which level-set ACMs are implemented as RNNs. Deep Active Contours by Rupprecht *et al.* [111], is another example. For medical image segmentation, Hatamizadeh *et al.* [112] proposed an integrated Deep Active Lesion Segmentation (DALS) model that trains the FCN backbone to predict the parameter functions of a novel, locally-parameterized level-set energy functional. In another relevant effort, Marcos *et al.* [113] proposed Deep Structured Active Contours (DSAC), which combines ACMs and pre-trained FCNs in a structured prediction framework for building instance segmentation (albeit with manual initialization) in aerial images. For the same application, Cheng *et al.* [114] proposed the Deep Active Ray Network (DarNet), which is similar to DSAC, but with a different explicit ACM formulation based on polar coordinates to prevent contour self-intersection. A truly end-to-end backpropagation trainable, fully-integrated FCN-ACM combination was recently introduced by Hatamizadeh *et al.* [115], dubbed Deep Convolutional Active Contours (DCAC).

3.11 Other Models

In addition to the above models, there are several other popular DL architectures for segmentation, such as the following: Context Encoding Network (EncNet) that uses a basic feature extractor and feeds the feature maps into a Context Encoding Module [116]. RefineNet [117], which is a multi-path refinement network that explicitly exploits all the information available along the down-sampling process to enable high-resolution prediction using long-range residual connections. Seednet [118], which introduced an automatic seed generation technique with deep reinforcement learning that learns to solve the interactive segmentation problem, Feedforward-Net [124] which maps image super-pixels to rich feature representations extracted from a sequence of nested regions of increasing extent and exploits statistical structures in the image and in the label space without setting up explicit structured prediction mechanisms.

Yet additional models include BoxSup [120], Graph convolutional networks [121], Wide ResNet [122], Exfuse (enhancing low-level and high-level features fusion) [123], dual image segmentation (DIS) [125], FoveaNet (Perspective-aware scene parsing) [126], Ladder DenseNet [127], Bilateral segmentation network (BiSeNet) [128], Semantic Prediction Guidance for Scene Parsing (SPGNet) [129], Gated shape CNNs [130], Adaptive context network (AC-Net) [131], Dynamic-structured semantic propagation network (DSSPN) [132], symbolic graph reasoning (SGR) [133], CascadeNet [134], Scale-adaptive convolutions (SAC) [135], Unified perceptual parsing (UperNet) [136].

Panoptic segmentation [137] is also another interesting (and newer) segmentation problem with rising popularity, and there are already several interesting works on this direction, including Panoptic Feature Pyramid Network [138], attention-guided network for Panoptic segmentation [139], and Seamless Scene Segmentation [140].

Figure 42 illustrates the timeline of popular DL-based works for semantic segmentation, as well as instance segmentation since 2014. Given the large number of works developed in the last few years, we only show some of the most representative ones.

4 Image Segmentation Datasets

In this section we provide a summary of some of the most widely used image segmentation datasets. We group these datasets into 3 categories—2D images, 2.5D RGB-D (color+depth) images, and 3D images—and provide details about the characteristics of each dataset. The listed datasets have pixel-wise labels, which can be used for evaluating model performance.

It is worth mentioning that some of these works, use **data augmentation** to increase the number of labeled samples, specially the ones which deal with small datasets (such as in medical domain). Data augmentation serves to increase the number of training samples by applying a set of transformation (either in the data space, or feature space, or sometimes both) to the images (i.e., both the input image and the segmentation map). Some typical transformations include translation, reflection, rotation, warping, scaling, color space shifting, cropping, and projections onto principal components. Data augmentation has proven to improve the performance of the models, especially when learning from limited datasets, such as those in medical image analysis. It can also be beneficial in yielding faster convergence, decreasing the chance of over-fitting, and enhancing generalization. For some small datasets, data augmentation has been shown to boost model performance more than 20%.

4.1 2D Datasets

The majority of image segmentation research has focused on 2D images; therefore, many 2D image segmentation datasets are available. The following are some of the most popular:

PASCAL Visual Object Classes (VOC) [141] is one of most popular datasets in computer vision, with annotated images available for 5 tasks—classification, segmentation, detection, action recognition, and person layout. Nearly all popular segmentation algorithms reported in the literature have been evaluated on this dataset. For the segmentation task, there are 21 classes of object labels—vehicles, household, animals, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, TV/monitor, bird, cat, cow, dog, horse, sheep, and person (pixel are labeled as background if they do not belong to any of these classes). This dataset is divided into two sets, training and validation, with 1,464 and 1,449 images, respectively. There is a private test set for the actual challenge. Figure 43 shows an example image and its pixel-wise label.

PASCAL Context [142] is an extension of the PASCAL VOC 2010 detection challenge, and it contains pixel-wise labels for all training images. It contains more than 400 classes (including the original 20 classes plus backgrounds from PASCAL VOC segmentation), divided into three categories (objects, stuff, and hybrids). Many of the object categories of this dataset are too sparse and; therefore, a subset of 59 frequent classes are usually selected for use. Figure 44 shows the segmentation map of three sample images of this dataset.

Microsoft Common Objects in Context (MS COCO) [143] is another large-scale object detection, segmentation, and captioning dataset. COCO includes images of complex everyday scenes, containing common objects in their natural contexts. This dataset contains photos of 91 objects types, with a total of 2.5 million labeled instances in 328k images. It has been used mainly for segmenting individual object instances. Figure 45 shows the difference between MS COCO labels and the previous datasets for a given sample image. The detection challenge includes more than 80 classes, providing more than 82k images for training, 40.5k images for validation, and more than 80k images for its test set.

Cityscapes [144] is a large-scale database with a focus on semantic understanding of urban street scenes. It contains a diverse set of stereo video sequences recorded in street scenes from 50 cities, with high quality pixel-level annotation of 5k frames, in addition to a set of 20k weakly annotated frames. It includes semantic and dense pixel annotations of 30 classes, grouped into 8 categories—flat surfaces, humans, vehicles, constructions, objects, nature, sky, and void. Figure 46 shows four sample segmentation maps from this dataset.

ADE20K/MIT Scene Parsing (SceneParse150) offers a standard training and evaluation platform for scene parsing algorithms. The data for this benchmark comes from the ADE20K dataset [134], which contains more than 20K scene-centric images exhaustively annotated with objects and object

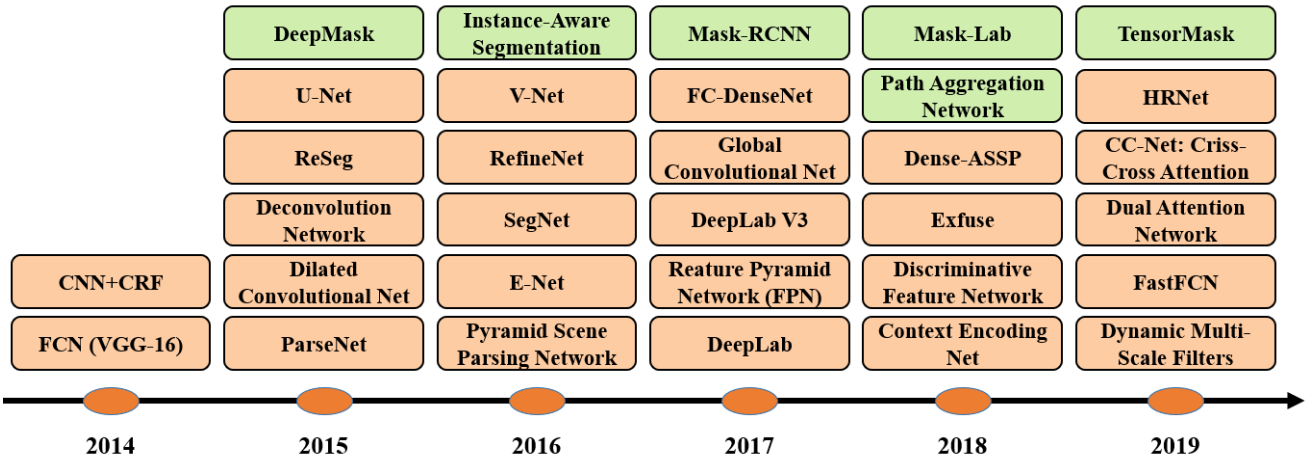


Fig. 42. The timeline of DL-based segmentation algorithms for 2D images. Orange and green blocks refer to semantic, and instance segmentation algorithms respectively.

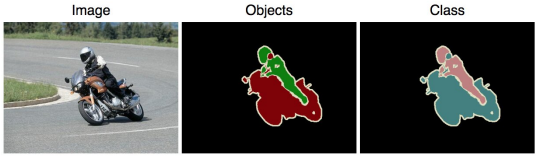


Fig. 43. An example image from the PASCAL VOC dataset. From [141].

Fig. 44. Three sample images and segmentation maps from the PASCAL context dataset. From [142].

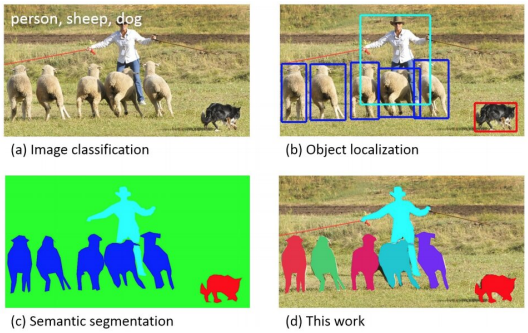


Fig. 45. A sample image and its segmentation map in COCO, and its comparison with previous datasets. From [143].

Fig. 46. Three sample images with their corresponding segmentation maps from the Cityscapes dataset. From [144].

parts. The benchmark is divided into 20K images for training, 2K images for validation, and another batch of images for testing. There are 150 semantic categories in this dataset.

SiftFlow [145] includes 2,688 annotated images from a subset of the LabelMe database. The 256×256 pixel images are based on 8 different outdoor scenes, among them streets, mountains, fields, beaches, and buildings. All images belong to one of 33 semantic classes.

Stanford background [146] contains outdoor images of scenes from existing datasets, such as LabelMe, MSRC, and PASCAL VOC. It contains 715 images with at least one foreground object. The dataset is pixel-wise annotated, and can be used for semantic scene understanding. Semantic and geometric labels for this dataset were obtained using Amazon's Mechanical Turk (AMT).

Berkeley Segmentation Dataset (BSD) [147] contains 12,000 hand-labeled segmentations of 1,000 Corel dataset images from 30 human subjects. It aims to provide an empirical basis for research on image segmentation and boundary detection. Half of the segmentations were obtained from presenting the subject a color image and the other half from presenting a grayscale image. The public benchmark based on this data consists of all of the grayscale and color segmentations for 300 images. The images are divided into a training set of 200 images and a test set of 100 images.

Youtube-Objects [148] contains videos collected from YouTube, which include objects from ten PASCAL VOC classes (aeroplane, bird, boat, car, cat, cow, dog, horse, motorbike, and train). The original dataset did not contain pixel-wise annotations (as it was originally developed for object detection, with weak annotations). However, Jain *et al.* [149] manually annotated a subset of 126 sequences, and then extracted a subset of frames to further generate semantic labels. In total, there are about 10,167 annotated 480x360 pixel frames available in this dataset.

KITTI [150] is one of the most popular datasets for mobile robotics and autonomous driving. It contains hours of

videos of traffic scenarios, recorded with a variety of sensor modalities (including high-resolution RGB, grayscale stereo cameras, and a 3D laser scanners). The original dataset does not contain ground truth for semantic segmentation, but researchers have manually annotated parts of the dataset for research purposes. For example, Alvarez *et al.* [151] generated ground truth for 323 images from the road detection challenge with 3 classes, road, vertical, and sky.

Other Datasets are available for image segmentation purposes too, such as **Semantic Boundaries Dataset (SBD)** [152], **PASCAL Part** [153], **SYNTHIA** [154], and **Adobes Portrait Segmentation** [155].

4.2 2.5D Datasets

With the availability of affordable range scanners, RGB-D images have became popular in both research and industrial applications. The following RGB-D datasets are some of the most popular:

NYU-D V2 [156] consists of video sequences from a variety of indoor scenes, recorded by the RGB and depth cameras of the Microsoft Kinect. It includes 1,449 densely labeled pairs of aligned RGB and depth images from more than 450 scenes taken from 3 cities. Each object is labeled with a class and an instance number (e.g., cup1, cup2, cup3, etc.). It also contains 407,024 unlabeled frames. This dataset is relatively small compared to other existing datasets. Figure 47 shows a sample image and its segmentation map.

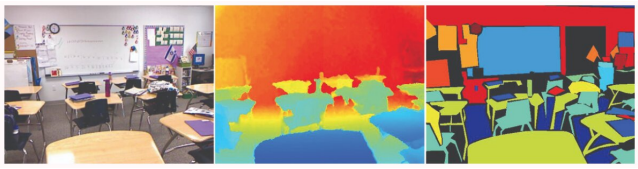


Fig. 47. A sample image from the NYU V2 dataset. From left: the RGB image, pre-processed depth, and set of labels. From [156].

SUN-3D [157] is a large-scale RGB-D video dataset that contains 415 sequences captured for 254 different spaces in 41 different buildings; 8 sequences are annotated and more will be annotated in the future. Each annotated frame comes with the semantic segmentation of the objects in the scene, as well as information about the camera pose.

SUN RGB-D [158] provides an RGB-D benchmark for the goal of advancing the state-of-the-art in all major scene understanding tasks. It is captured by four different sensors and contains 10,000 RGB-D images at a scale similar to PASCAL VOC. The whole dataset is densely annotated and includes 146,617 2D polygons and 58,657 3D bounding boxes with accurate object orientations, as well as the 3D room category and layout for scenes. Figure 48 shows two example images (with annotations).

UW RGB-D Object Dataset [159] contains 300 common household objects recorded using a Kinect style 3D camera. The objects are organized into 51 categories, arranged using WordNet hypernym-hyponym relationships (similar to ImageNet). This dataset was recorded using a Kinect style 3D camera that records synchronized and aligned 640×480 pixel RGB and depth images at 30 Hz. This dataset also includes 8 annotated video sequences of natural scenes, containing objects from the dataset (the UW RGB-D Scenes Dataset).

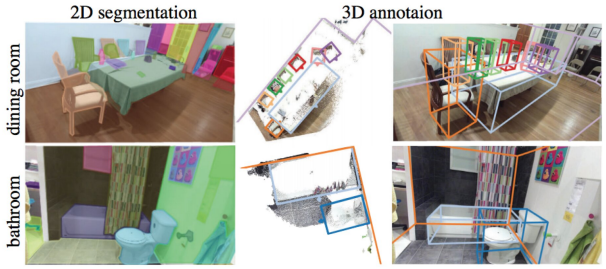


Fig. 48. Two example images (with annotations) from SUN RGB-D dataset. From [158].

ScanNet [160] is an RGB-D video dataset containing 2.5 million views in more than 1,500 scans, annotated with 3D camera poses, surface reconstructions, and instance-level semantic segmentations. To collect these data, an easy-to-use and scalable RGB-D capture system was designed that includes automated surface reconstruction, and the semantic annotation was crowd-sourced. Using this data helped achieve state-of-the-art performance on several 3D scene understanding tasks, including 3D object classification, semantic voxel labeling, and CAD model retrieval.

4.3 3D Datasets

3D image datasets are popular in robotic, medical image analysis, 3D scene analysis, and construction applications. Three dimensional images are usually provided via meshes or other volumetric representations, such as point clouds. Here, we mention some of the popular 3D datasets.

Stanford 2D-3D: This dataset provides a variety of mutually registered modalities from 2D, 2.5D and 3D domains, with instance-level semantic and geometric annotations [161], and is collected in 6 indoor areas. It contains over 70,000 RGB images, along with the corresponding depths, surface normals, semantic annotations, global XYZ images as well as camera information.

ShapeNet Core: ShapeNetCore is a subset of the full ShapeNet dataset [162] with single clean 3D models and manually verified category and alignment annotations [163]. It covers 55 common object categories with about 51,300 unique 3D models.

Sydney Urban Objects Dataset: This dataset contains a variety of common urban road objects, collected in the central business district of Sydney, Australia. There are 631 individual scans of objects across classes of vehicles, pedestrians, signs and trees [164].

5 Performance Review

In this section, we first provide a summary of some of the popular metrics used in evaluating the performance of segmentation models, and then we provide the quantitative performance of the promising DL-based segmentation models on popular datasets.

5.1 Metrics For Segmentation Models

Ideally, a model should be evaluated in multiple respects, such as quantitative accuracy, speed (inference time), and

storage requirements (memory footprint). Measuring speed can be tricky, as it depends on the hardware and experimental conditions, but it is an important factor in real-time applications, as is the memory footprint if a model is intended for small devices with limited memory capacity. However, most of the research works so far, focus on the metrics for evaluating the model accuracy. Below we summarize the most popular metrics for assessing the accuracy of segmentation algorithms. Although quantitative metrics are used to compare different models on benchmarks, the visual quality of model outputs is also important in deciding which model is best (as human is the final consumer of many of the models developed for computer vision applications).

Pixel accuracy simply finds the ratio of pixels properly classified, divided by the total number of pixels. For $K+1$ classes (K foreground classes and the background) pixel accuracy is defined as Eq 2:

$$\text{PA} = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K}\sum_{j=0}^{K} p_{ij}}, \tag{2}$$

where p_{ij} is the number of pixels of class i predicted as belonging to class j.

Mean Pixel Accuracy (MPA) is the extended version of PA, in which the ratio of correct pixels is computed in a per-class manner and then averaged over the total number of classes, as in Eq 3:

$$\text{MPA} = \frac{1}{K+1}\sum_{i=0}^{K}\frac{p_{ii}}{\sum_{j=0}^{K} p_{ij}}. \tag{3}$$

Intersection over Union (IoU) or the **Jaccard Index** is one of the most commonly used metrics in semantic segmentation. It is defined as the area of intersection between the predicted segmentation map and the ground truth, divided by the area of union between the predicted segmentation map and the ground truth:

$$\text{IoU} = J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \tag{4}$$

where A and B denote the ground truth and the predicted segmentation maps, respectively. It ranges between 0 and 1.

Mean-IoU is another popular metric, which is defined as the average IoU over all classes. It is widely used in reporting the performance of modern segmentation algorithms.

Precision / Recall / F1 score are popular metrics for reporting the accuracy of many of the classical image segmentation models. Precision and recall can be defined for each class, as well as at the aggregate level, as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad \text{Recall} \quad = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{5}$$

where TP refers to the true positive fraction, FP refers to the false positive fraction, and FN refers to the false negative fraction. Usually we are interested into a combined version of precision and recall rates. A popular such a metric is called the F1 score, which is defined as the harmonic mean of precision and recall:

$$\text{F1-score} = \frac{2\ \text{Prec}\ \text{Rec}}{\text{Prec}+\text{Rec}}. \tag{6}$$

Dice coefficient is another popular metric for image segmentation, which can be defined as twice the overlap area of predicted and ground-truth maps, divided by the total number of pixels in both images. The Dice coefficient is very similar to the IoU:

$$\text{Dice} = \frac{2|A \cap B|}{|A|+|B|}. \tag{7}$$

When applied to boolean data (e.g., binary segmentation maps), and referring to the foreground as a positive class, the Dice coefficient is essentially identical to the F1 score, defined as Eq 8:

$$\text{Dice} = \frac{2\text{TP}}{2\text{TP}+\text{FP}+\text{FN}} = \text{F1}. \tag{8}$$

The Dice coefficient and IoU are positively correlated.

5.2 Quantitative Performance of DL-Based Models

In this section we tabulate the performance of several of the previously discussed algorithms on popular segmentation benchmarks. It is worth mentioning that although most models report their performance on standard datasets and use standard metrics, some of them fail to do so, making across-the-board comparisons difficult. Furthermore, only a small percentage of publications provide additional information, such as execution time and memory footprint, in a reproducible way, which is important to industrial applications of segmentation models (such as drones, self-driving cars, robotics, etc.) that may run on embedded consumer devices with limited computational power and storage, making fast, light-weight models crucial.

The following tables summarize the performances of several of the prominent DL-based segmentation models on different datasets. Table 1 focuses on the PASCAL VOC test set. Clearly, there has been much improvement in the accuracy of the models since the introduction of the FCN, the first DL-based image segmentation model.[1] Table 2 focuses on the Cityscape test dataset. The latest models feature about 23% relative gain over the initial FCN model on this dataset. Table 3 focuses on the MS COCO stuff test set. This dataset is more challenging than PASCAL VOC, and Cityescapes, as the highest mIoU is approximately 40%. Table 4 focuses on the ADE20k validation set. This dataset is also more challenging than the PASCAL VOC and Cityescapes datasets. Finally, Table 5 summarizes the performance of several prominent models for RGB-D segmentation on the NYUD-v2 and SUN-RGBD datasets.

To summarize the tabulated data, there has been significant progress in the performance of deep segmentation models over the past 5–6 years, with a relative improvement of 25%-42% in mIoU on different datasets. However, some publications suffer from lack of reproducibility for multiple reasons—they report performance on non-standard benchmarks/databases, or they report performance only on arbitrary subsets of the test set from a popular benchmark, or they do not adequately describe the experimental setup and sometimes evaluate the model performance only on a subset of object classes. Most importantly, many publications do not

1. Note that some works report two versions of their models: one which is only trained on PASCAL VOC and another that is pre-trained on a different dataset (such as MS-COCO, ImageNet, or JFT-300M) and then fine-tuned on VOC.

TABLE 1
Accuracies of segmentation models on the PASCAL VOC test set. (* Refers to the model pre-trained on another dataset, such as MS-COCO, ImageNet, or JFT-300M.)

Method	Backbone	mIoU
FCN [32]	VGG-16	62.2
CRF-RNN [40]	-	72.0
CRF-RNN* [40]	-	74.7
BoxSup* [120]	-	75.1
Piecewise [41]	-	75.3
Piecewise* [41]	-	78.0
DPN [42]	-	74.1
DPN* [42]	-	77.5
DeepLab-CRF [78]	ResNet-101	79.7
GCN* [121]	ResNet-152	82.2
RefineNet [117]	ResNet-152	84.2
Wide ResNet [122]	WideResNet-38	84.9
OCR [119]	ResNet-101	84.3
OCR [57]	HRNetV2-W48	84.5
PSPNet [57]	ResNet-101	85.4
DeeplabV3 [12]	ResNet-101	85.7
PSANet [98]	ResNet-101	85.7
EncNet [116]	ResNet-101	85.9
DFN [99]	ResNet-101	82.7
DFN* [99]	ResNet-101	86.2
Exfuse [123]	ResNet-101	86.2
SDN [46]	DenseNet-161	83.5
SDN* [46]	DenseNet-161	86.6
DIS [125]	ResNet-101	86.8
DM-Net [59]	ResNet-101	84.4
DM-Net* [59]	ResNet-101	87.06
APC-Net [61]	ResNet-101	84.2
APC-Net* [61]	ResNet-101	87.1
EMANet [95]	ResNet-101	87.7
DeeplabV3+ [83]	Xception-71	87.8
Exfuse [123]	ResNeXt-131	87.9
MSCI [62]	ResNet-152	88.0
EMANet [95]	ResNet-152	88.2
DeeplabV3+* [83]	Xception-71	89.0

TABLE 2
Accuracies of segmentation models on the Cityescapes dataset.

Method	Backbone	mIoU
SegNet basic [44]	-	57.0
FCN-8s [32]	-	65.3
DPN [42]	-	66.8
Dilation10 [79]	-	67.1
DeeplabV2 [78]	ResNet-101	70.4
RefineNet [117]	ResNet-101	73.6
FoveaNet [126]	ResNet-101	74.1
Ladder DenseNet [127]	Ladder DenseNet-169	73.7
GCN [121]	ResNet-101	76.9
DUC-HDC [80]	ResNet-101	77.6
Wide ResNet [122]	WideResNet-38	78.4
PSPNet [57]	ResNet-101	85.4
BiSeNet [128]	ResNet-101	78.9
DFN [99]	ResNet-101	79.3
PSANet [98]	ResNet-101	80.1
DenseASPP [81]	DenseNet-161	80.6
SPGNet [129]	2xResNet-50	81.1
DANet [93]	ResNet-101	81.5
CCNet [96]	ResNet-101	81.4
DeeplabV3 [12]	ResNet-101	81.3
DeeplabV3 [83]	Xception-71	82.1
AC-Net [131]	ResNet-101	82.3
OCR [119]	ResNet-101	82.4
GS-CNN [130]	WideResNet	82.8
HRNetV2+OCR (w/ASPP) [119]	HRNetV2-W48	83.7

TABLE 3
Accuracies of segmentation models on the MS COCO stuff dataset.

Method	Backbone	mIoU
RefineNet [117]	ResNet-101	33.6
CCN [60]	Ladder DenseNet-101	35.7
DANet [93]	ResNet-50	37.9
DSSPN [132]	ResNet-101	37.3
EMA-Net [95]	ResNet-50	37.5
SGR [133]	ResNet-101	39.1
OCR [119]	ResNet-101	39.5
DANet [93]	ResNet-101	39.7
EMA-Net [95]	ResNet-50	39.9
AC-Net [131]	ResNet-101	40.1
OCR [119]	HRNetV2-W48	40.5

TABLE 4
Accuracies of segmentation models on the ADE20k validation dataset.

Method	Backbone	mIoU
FCN [32]	-	29.39
DilatedNet [79]	-	32.31
CascadeNet [134]	-	34.9
RefineNet [117]	ResNet-152	40.7
PSPNet [57]	ResNet-101	43.29
PSPNet [57]	ResNet-269	44.94
EncNet [116]	ResNet-101	44.64
SAC [135]	ResNet-101	44.3
PSANet [98]	ResNet-101	43.7
UperNet [136]	ResNet-101	42.66
DSSPN [132]	ResNet-101	43.68
DM-Net [59]	ResNet-101	45.5
OCR [119]	HRNetV2-W48	45.6
AC-Net [131]	ResNet-101	45.9

provide the source-code for their model implementations. However, with the increasing popularity of deep learning models, the trend has been positive and many research groups are moving toward reproducible frameworks and open-sourcing their implementations.

6 Challenges and Opportunities

There is not doubt that image segmentation has benefited greatly from deep learning, but several challenges lie ahead. We will next introduce some of the promising research

TABLE 5
Performance of segmentation models on the NYUD-v2, and SUN-RGBD datasets, in terms of mIoU, and mean Accuracy (mAcc).

	NYUD-v2		SUN-RGBD	
Method	m-Acc	m-IoU	m-Acc	m-IoU
Mutex [165]	-	31.5	-	-
MS-CNN [166]	45.1	34.1	-	-
FCN [32]	46.1	34.0	-	-
Joint-Seg [167]	52.3	39.2	-	-
SegNet [44]	-	-	44.76	31.84
Structured Net [41]	53.6	40.6	53.4	42.3
B-SegNet [45]	-	-	45.9	30.7
3D-GNN [168]	55.7	43.1	57.0	45.9
LSD-Net [49]	60.7	45.9	58.0	-
RefineNet [117]	58.9	46.5	58.5	45.9
D-aware CNN [169]	61.1	48.4	53.5	42.0
RDFNet [170]	62.8	50.1	60.1	47.7
G-Aware Net [171]	68.7	59.6	74.9	54.5

directions that we believe will help in further advancing image segmentation algorithms.

6.1 More Challenging Datasets

Several large-scale image datasets have been created for semantic segmentation and instance segmentation. However, there remains a need for more challenging datasets, as well as datasets for different kinds of images. For still images, datasets with a large number of objects and overlapping objects would be very valuable. This can enable training models that are better at handling dense object scenarios, as well as large overlaps among objects as is common in real-world scenarios.

With the rising popularity of 3D image segmentation, especially in medical image analysis, there is also a strong need for large-scale 3D images datasets. These datasets are more difficult to create than their lower dimensional counterparts. Existing datasets for 3D image segmentation available are typically not large enough, and some are synthetic, and therefore larger and more challenging 3D image datasets can be very valuable.

6.2 Interpretable Deep Models

While DL-based models have achieved promising performance on challenging benchmarks, there remain open questions about these models. For example, what exactly are deep models learning? How should we interpret the features learned by these models? What is a minimal neural architecture that can achieve a certain segmentation accuracy on a given dataset? Although some techniques are available to visualize the learned convolutional kernels of these models, a concrete study of the underlying behavior/dynamics of these models is lacking. A better understanding of the theoretical aspects of these models can enable the development of better models curated toward various segmentation scenarios.

6.3 Weakly-Supervised and Unsupervised Learning

Weakly-supervised (a.k.a. few shot learning) and unsupervised learning are becoming very active research areas. These techniques promise to be specially valuable for image segmentation, as collecting labeled samples for segmentation problem is problematic in many application domains, particularly so in medical image analysis. The transfer learning approach is to train a generic image segmentation model on a large set of labeled samples (perhaps from a public benchmark), and then fine-tune that model on a few samples from some specific target application. Self-supervised learning is another promising direction that is attracting much attraction in various fields. There are many details in images that that can be captured to train a segmentation models with far fewer training samples, with the help of self-supervised learning. Models based on reinforcement learning could also be another potential future direction, as they have scarcely received attention for image segmentation.

6.4 Real-time Models for Various Applications

In many applications, accuracy is the most important factor; however, there are applications in which it is also critical to have segmentation models that can run in near real-time, or at least near common camera frame rates (at least 25 frames per second). This is useful for computer vision systems that are, for example, deployed in autonomous vehicles. Most of the current models are far from this frame-rate; e.g., FCN-8 takes roughly 100 ms to process a low-resolution image. Models based on dilated convolution help to increase the speed of segmentation models to some extent, but there is still plenty of room for improvement.

6.5 Memory Efficient Models

Many modern segmentation models require a significant amount of memory even during the inference stage. So far, much effort has been directed towards improving the accuracy of such models, but in order to fit them into specific devices, such as mobile phones, the networks must be simplified. This can be done either by using simpler models, or by using model compression techniques, or even training a complex model and then using knowledge distillation techniques to compress it into a smaller, memory efficient network that mimics the complex model.

6.6 3D Point-Cloud Segmentation

Numerous works have focused on 2D image segmentation, but much fewer have addressed 3D point-cloud segmentation. However, there has been an increasing interest in point-cloud segmentation, which has a wide range of applications, in 3D modeling, self-driving cars, robotics, building modeling, etc. Dealing with 3D unordered and unstructured data such as point clouds poses several challenges. For example, the best way to apply CNNs and other classical deep learning architectures on point clouds is unclear. Graph-based deep models can be a potential area of exploration for point-cloud segmentation, enabling additional industrial applications of these data.

7 Conclusions

We have surveyed more than 100 recent image segmentation algorithms based on deep learning models, which have achieved impressive performance in various image segmentation tasks and benchmarks, grouped into ten categories such as: CNN and FCN, RNN, R-CNN, dilated CNN, attention-based models, generative and adversarial models, among others. We summarized quantitative performance analyses of these models on some popular benchmarks, such as the PASCAL VOC, MS COCO, Cityscapes, and ADE20k datasets. Finally, we discussed some of the open challenges and potential research directions for image segmentation that could be pursued in the coming years.

Acknowledgments

The authors would like to thank Tsung-Yi Lin from Google Brain, and Jingdong Wang and Yuhui Yuan from Microsoft Research Asia, for reviewing this work, and providing very helpful comments and suggestions.

REFERENCES

[1] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[2] D. Forsyth and J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[3] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[4] R. Nock and F. Nielsen, "Statistical region merging," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 26, no. 11, pp. 1452–1458, 2004.

[5] N. Dhanachandra, K. Manglem, and Y. J. Chanu, "Image segmentation using k-means clustering algorithm and subtractive clustering algorithm," *Procedia Computer Science*, vol. 54, pp. 764–771, 2015.

[6] L. Najman and M. Schmitt, "Watershed of a continuous function," *Signal Processing*, vol. 38, no. 1, pp. 99–112, 1994.

[7] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[8] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[9] N. Plath, M. Toussaint, and S. Nakajima, "Multi-class image segmentation using conditional random fields and global classification," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 817–824.

[10] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE transactions on image processing*, vol. 14, no. 10, pp. 1570–1582, 2005.

[11] S. Minaee and Y. Wang, "An admm approach to masked signal decomposition using subspace representation," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3192–3204, 2019.

[12] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[13] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[17] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[18] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE transactions on acoustics, speech, and signal processing*, vol. 37, no. 3, pp. 328–339, 1989.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[23] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[25] D. E. Rumelhart, G. E. Hinton, R. J. Williams *et al.*, "Learning representations by back-propagating errors," *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.

[26] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[28] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[29] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[30] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[31] https://github.com/hindupuravinash/the-gan-zoo.

[32] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[33] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[34] G. Wang, W. Li, S. Ourselin, and T. Vercauteren, "Automatic brain tumor segmentation using cascaded anisotropic convolutional neural networks," in *International MICCAI Brainlesion Workshop*. Springer, 2017, pp. 178–190.

[35] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2359–2367.

[36] Y. Yuan, M. Chao, and Y.-C. Lo, "Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance," *IEEE transactions on medical imaging*, vol. 36, no. 9, pp. 1876–1886, 2017.

[37] N. Liu, H. Li, M. Zhang, J. Liu, Z. Sun, and T. Tan, "Accurate iris segmentation in non-cooperative environments using fully convolutional networks," in *2016 International Conference on Biometrics (ICB)*. IEEE, 2016, pp. 1–8.

[38] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[39] A. G. Schwing and R. Urtasun, "Fully connected deep structured networks," *arXiv preprint arXiv:1503.02351*, 2015.

[40] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1529–1537.

[41] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3194–3203.

[42] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1377–1385.

[43] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520–1528.

[44] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[45] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[46] J. Fu, J. Liu, Y. Wang, J. Zhou, C. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," *IEEE Transactions on Image Processing*, 2019.

[47] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[48] X. Xia and B. Kulis, "W-net: A deep model for fully unsupervised image segmentation," *arXiv preprint arXiv:1711.08506*, 2017.

[49] Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang, "Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor

semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3029–3037.

[50] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[51] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 565–571.

[52] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 424–432.

[53] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.

[54] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[55] T. Brosch, L. Y. Tang, Y. Yoo, D. K. Li, A. Traboulsee, and R. Tam, "Deep 3d convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1229–1239, 2016.

[56] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[57] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[58] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 519–534.

[59] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3562–3572.

[60] H. Ding, X. Jiang, B. Shuai, A. Qun Liu, and G. Wang, "Context contrasted feature and gated multi-scale aggregation for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2393–2402.

[61] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7519–7528.

[62] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 603–619.

[63] G. Li, Y. Xie, L. Lin, and Y. Yu, "Instance-level salient object segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2386–2395.

[64] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[65] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[66] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[67] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3150–3158.

[68] R. Hu, P. Dollár, K. He, T. Darrell, and R. Girshick, "Learning to segment every thing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4233–4241.

[69] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "Masklab: Instance segmentation by refining object detection with semantic and direction features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4013–4022.

[70] X. Chen, R. Girshick, K. He, and P. Dollár, "Tensormask: A foundation for dense object segmentation," *arXiv preprint arXiv:1903.12174*, 2019.

[71] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[72] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 2015, pp. 1990–1998.

[73] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*. Springer, 2016, pp. 75–91.

[74] E. Xie, P. Sun, X. Song, W. Wang, X. Liu, D. Liang, C. Shen, and P. Luo, "Polarmask: Single shot instance segmentation with polar representation," *arXiv preprint arXiv:1909.13226*, 2019.

[75] Z. Hayder, X. He, and M. Salzmann, "Boundary-aware instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5696–5704.

[76] M. Bai and R. Urtasun, "Deep watershed transform for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5221–5229.

[77] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy, "Semantic instance segmentation via deep metric learning," *arXiv preprint arXiv:1703.10277*, 2017.

[78] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[79] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[80] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *winter conference on applications of computer vision*. IEEE, 2018, pp. 1451–1460.

[81] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "Denseaspp for semantic segmentation in street scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3684–3692.

[82] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[83] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[84] F. Visin, M. Ciccone, A. Romero, K. Kastner, K. Cho, Y. Bengio, M. Matteucci, and A. Courville, "Reseg: A recurrent neural network-based model for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 41–48.

[85] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville, and Y. Bengio, "Renet: A recurrent neural network based alternative to convolutional networks," *arXiv preprint arXiv:1505.00393*, 2015.

[86] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki, "Scene labeling with lstm recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3547–3555.

[87] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph lstm," in *European Conference on Computer Vision*. Springer, 2016, pp. 125–143.

[88] Y. Xiang and D. Fox, "Da-rnn: Semantic mapping with data associated recurrent neural networks," *arXiv preprint arXiv:1703.03098*, 2017.

[89] R. Hu, M. Rohrbach, and T. Darrell, "Segmentation from natural language expressions," in *European Conference on Computer Vision*. Springer, 2016, pp. 108–124.

[90] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3640–3649.

[91] Q. Huang, C. Xia, C. Wu, S. Li, Y. Wang, Y. Song, and C.-C. J. Kuo, "Semantic segmentation with reverse attention," *arXiv preprint arXiv:1707.06426*, 2017.

[92] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," *arXiv preprint arXiv:1805.10180*, 2018.

[93] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.

[94] Y. Yuan and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916*, 2018.

[95] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9167–9176.

[96] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 603–612.

[97] M. Ren and R. S. Zemel, "End-to-end instance segmentation with recurrent attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6656–6664.

[98] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. Change Loy, D. Lin, and J. Jia, "Psanet: Point-wise spatial attention network for scene parsing," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 267–283.

[99] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1857–1866.

[100] P. Luc, C. Couprie, S. Chintala, and J. Verbeek, "Semantic segmentation using adversarial networks," *arXiv preprint arXiv:1611.08408*, 2016.

[101] N. Souly, C. Spampinato, and M. Shah, "Semi supervised semantic segmentation using generative adversarial network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5688–5696.

[102] W.-C. Hung, Y.-H. Tsai, Y.-T. Liou, Y.-Y. Lin, and M.-H. Yang, "Adversarial learning for semi-supervised semantic segmentation," *arXiv preprint arXiv:1802.07934*, 2018.

[103] Y. Xue, T. Xu, H. Zhang, L. R. Long, and X. Huang, "Segan: Adversarial network with multi-scale l 1 loss for medical image segmentation," *Neuroinformatics*, vol. 16, no. 3-4, pp. 383–392, 2018.

[104] M. Majurski, P. Manescu, S. Padi, N. Schaub, N. Hotaling, C. Simon Jr, and P. Bajcsy, "Cell image segmentation using generative adversarial networks, transfer learning, and augmentations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[105] K. Ehsani, R. Mottaghi, and A. Farhadi, "Segan: Segmenting and generating the invisible," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6144–6153.

[106] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[107] X. Chen, B. M. Williams, S. R. Vallabhaneni, G. Czanner, R. Williams, and Y. Zheng, "Learning active contour models for medical image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 632–11 640.

[108] S. Gur, L. Wolf, L. Golgher, and P. Blinder, "Unsupervised microvascular image segmentation using an active contours mimicking neural network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 10 722–10 731.

[109] P. Marquez-Neila, L. Baumela, and L. Alvarez, "A morphological approach to curvature-based evolution of curves and surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 2–17, 2014.

[110] T. H. N. Le, K. G. Quach, K. Luu, C. N. Duong, and M. Savvides, "Reformulating level sets as deep recurrent neural network approach to semantic segmentation," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2393–2407, 2018.

[111] C. Rupprecht, E. Huaroc, M. Baust, and N. Navab, "Deep active contours," *arXiv preprint arXiv:1607.05074*, 2016.

[112] A. Hatamizadeh, A. Hoogi, D. Sengupta, W. Lu, B. Wilcox, D. Rubin, and D. Terzopoulos, "Deep active lesion segmentation," in *Proc. International Workshop on Machine Learning in Medical Imaging*, ser. Lecture Notes in Computer Science, vol. 11861. Springer, 2019, pp. 98–105.

[113] D. Marcos, D. Tuia, B. Kellenberger, L. Zhang, M. Bai, R. Liao, and R. Urtasun, "Learning deep structured active contours end-to-end," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8877–8885.

[114] D. Cheng, R. Liao, S. Fidler, and R. Urtasun, "Darnet: Deep active ray network for building segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7431–7439.

[115] A. Hatamizadeh, D. Sengupta, and D. Terzopoulos, "End-to-end deep convolutional active contours for image segmentation," *arXiv preprint arXiv:1909.13359*, 2019.

[116] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7151–7160.

[117] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[118] G. Song, H. Myeong, and K. Mu Lee, "Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1760–1768.

[119] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," *arXiv preprint arXiv:1909.11065*, 2019.

[120] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1635–1643.

[121] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4353–4361.

[122] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the resnet model for visual recognition," *Pattern Recognition*, vol. 90, pp. 119–133, 2019.

[123] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 269–284.

[124] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3376–3385.

[125] P. Luo, G. Wang, L. Lin, and X. Wang, "Deep dual learning for semantic image segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2718–2726.

[126] X. Li, Z. Jie, W. Wang, C. Liu, J. Yang, X. Shen, Z. Lin, Q. Chen, S. Yan, and J. Feng, "Foveanet: Perspective-aware urban scene parsing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 784–792.

[127] I. Kreso, S. Segvic, and J. Krapac, "Ladder-style densenets for semantic segmentation of large natural images," in *IEEE International Conference on Computer Vision*, 2017, pp. 238–245.

[128] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *European Conference on Computer Vision*, 2018, pp. 325–341.

[129] B. Cheng, L.-C. Chen, Y. Wei, Y. Zhu, Z. Huang, J. Xiong, T. S. Huang, W.-M. Hwu, and H. Shi, "Spgnet: Semantic prediction guidance for scene parsing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5218–5228.

[130] T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, "Gated-scnn: Gated shape cnns for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2019, pp. 5229–5238.

[131] J. Fu, J. Liu, Y. Wang, Y. Li, Y. Bao, J. Tang, and H. Lu, "Adaptive context network for scene parsing," in *Proceedings of the IEEE international conference on computer vision*, 2019, pp. 6748–6757.

[132] X. Liang, H. Zhou, and E. Xing, "Dynamic-structured semantic propagation network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 752–761.

[133] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing, "Symbolic graph reasoning meets convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 1853–1863.

[134] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[135] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan, "Scale-adaptive convolutions for scene parsing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2031–2039.

[136] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 418–434.

[137] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.

[138] A. Kirillov, R. Girshick, K. He, and P. Dollar, "Panoptic feature pyramid networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6399–6408.

[139] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang, "Attention-guided unified network for panoptic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7026–7035.

[140] L. Porzi, S. R. Bulo, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8277–8286.

[141] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[142] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 891–898.

[143] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014.

[144] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[145] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[146] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 1–8.

[147] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[148] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3282–3289.

[149] S. D. Jain and K. Grauman, "Supervoxel-consistent foreground propagation in video," in *European conference on computer vision*. Springer, 2014, pp. 656–671.

[150] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[151] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *European Conference on Computer Vision*. Springer, 2012, pp. 376–389.

[152] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 991–998.

[153] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1971–1978.

[154] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.

[155] X. Shen, A. Hertzmann, J. Jia, S. Paris, B. Price, E. Shechtman, and I. Sachs, "Automatic portrait segmentation for image stylization," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 93–102.

[156] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.

[157] J. Xiao, A. Owens, and A. Torralba, "Sun3d: A database of big spaces reconstructed using sfm and object labels," in *IEEE International Conference on Computer Vision*, 2013, pp. 1625–1632.

[158] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *IEEE conference on computer vision and pattern recognition*, 2015, pp. 567–576.

[159] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1817–1824.

[160] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5828–5839.

[161] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese, "Joint 2D-3D-Semantic Data for Indoor Scene Understanding," *ArXiv e-prints*, Feb. 2017.

[162] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[163] L. Yi, L. Shao, M. Savva, H. Huang, Y. Zhou, Q. Wang, B. Graham, M. Engelcke, R. Klokov, V. Lempitsky *et al.*, "Large-scale 3d shape reconstruction and segmentation from shapenet core55," *arXiv preprint arXiv:1710.06104*, 2017.

[164] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, "Unsupervised feature learning for classification of outdoor 3d scans," in *Australasian Conference on Robitics and Automation*, vol. 2, 2013, p. 1.

[165] Z. Deng, S. Todorovic, and L. Jan Latecki, "Semantic segmentation of rgbd images with mutex constraints," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1733–1741.

[166] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *IEEE international conference on computer vision*, 2015, pp. 2650–2658.

[167] A. Mousavian, H. Pirsiavash, and J. Kosecka, "Joint semantic segmentation and depth estimation with deep convolutional networks," in *International Conference on 3D Vision*. IEEE, 2016.

[168] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3d graph neural networks for rgbd semantic segmentation," in *IEEE International Conference on Computer Vision*, 2017, pp. 5199–5208.

[169] W. Wang and U. Neumann, "Depth-aware cnn for rgb-d segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 135–150.

[170] S.-J. Park, K.-S. Hong, and S. Lee, "Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation," in *IEEE International Conference on Computer Vision*, 2017, pp. 4980–4989.

[171] J. Jiao, Y. Wei, Z. Jie, H. Shi, R. W. Lau, and T. S. Huang, "Geometry-aware distillation for indoor semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2869–2878.

[172] V. Goel, J. Weng, and P. Poupart, "Unsupervised video object segmentation for deep reinforcement learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 5683–5694.

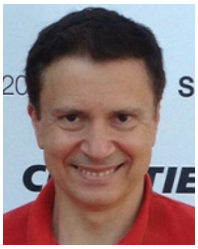
Shervin Minaee is a machine learning researcher at Snap Inc, where he works as a part of lens team. He received his PhD in Electrical Engineering and Computer Science from New York University. His research interest includes computer vision, image segmentation, biometrics recognition, and unsupervised learning, as well as natural language processing. He has published more than 40 papers and patents during his PhD. He has also previously worked as a data scientist at Expedia group, and as a research scientist at Samsung Research, AT&T Labs, and Huawei. He is a reviewer for more than 20 computer vision related journals from IEEE, ACM, and Elsevier, including IEEE Transactions on Image Processing, and International Journal of Computer Vision.

Yuri Boykov is a Professor at School of Computer Science at the University of Waterloo. His research is focused in the area of computer vision and biomedical image analysis with focus on modeling and optimization for structured segmentation, restoration, registration, stereo, motion, model fitting, recognition, photo-video editing and other data analysis problems. He is an editor for the International Journal of Computer Vision (IJCV). His work was listed among 10 most influential papers in IEEE Transactions of Pattern Analysis and Machine Intelligence (TPAMI Top Picks for 30 years). In 2017 Google Scholar listed his work on segmentation as a "classic paper in computer vision and pattern recognition". In 2011 he received Helmholtz Prize from IEEE and Test of Time Award by the International Conference on Computer Vision. The Faculty of Science at the University of Western Ontario recognized his work by awarding Distinguished Research Professorship in 2014 and Florence Bucke Prize in 2008.

Fatih Porikli is an IEEE Fellow and a Professor in the Research School of Engineering, Australian National University. He is acting as the Chief Scientist at Huawei, Santa Clara. He received his Ph.D. from New York University in 2002. His research interests include computer vision, pattern recognition, manifold learning, image enhancement, robust and sparse optimization and online learning with commercial applications in video surveillance, car navigation, intelligent transportation, satellite, and medical systems.

Antonio Plaza is a professor at the Department of Technology of Computers and Communications, University of Extremadura, where he received the M.Sc. degree in 1999 and the PhD degree in 2002, both in Computer Engineering. He has authored more than 600 publications, including 263 JCR journal papers (more than 170 in IEEE journals), 24 book chapters, and over 300 peer-reviewed conference proceeding papers. Prof. Plaza is a Fellow of IEEE for contributions to hyperspectral data processing and parallel computing of Earth observation data. He is a recipient of the recognition of Best Reviewers of the IEEE Geoscience and Remote Sensing Letters (in 2009) and a recipient of the recognition of Best Reviewers of the IEEE Transactions on Geoscience and Remote Sensing (in 2010), for which he served as Associate Editor in 2007-2012. He is a recipient of the Best Column Award of the IEEE Signal Processing Magazine in 2015, the 2013 Best Paper Award of the JSTARS journal, and the most highly cited paper (2005-2010) in the Journal of Parallel and Distributed Computing. He is included in the 2018 and 2019 Highly Cited Researchers List.

Nasser Kehtarnavaz is a Distinguished Professor at the Department of Electrical and Computer Engineering at the University of Texas at Dallas, Richardson, TX. His research interests include signal and image processing, machine learning, and real-time implementation on embedded processors. He has authored or co-authored ten books and more than 390 journal papers, conference papers, patents, manuals, and editorials in these areas. He is a Fellow of SPIE, a licensed Professional Engineer, and Editor-in-Chief of Journal of Real-Time Image Processing.



Demetri Terzopoulos is a Distinguished Professor of Computer Science at the University of California, Los Angeles, where he directs the UCLA Computer Graphics & Vision Laboratory. He is also Co-Founder and Chief Scientist of VoxelCloud, Inc. He graduated from McGill University in Honours Electrical Engineering and received the PhD degree in Artificial Intelligence from the Massachusetts Institute of Technology (MIT) in 1984. He is or was a Guggenheim Fellow, a Fellow of the ACM, IEEE, Royal Society of Canada, and Royal Society of London, and a Member of the European Academy of Sciences, the New York Academy of Sciences, and Sigma Xi. Among his many awards are an Academy Award from the Academy of Motion Picture Arts and Sciences for his pioneering work on physics-based computer animation, and the inaugural Computer Vision Distinguished Researcher Award from the IEEE for his pioneering and sustained research on deformable models and their applications. ISI and other indexes have listed him among the most highly-cited authors in engineering and computer science, with more than 400 published research papers and several volumes.He has given over 500 invited talks around the world about his research, including more than 100 distinguished lectures and keynote/plenary addresses. He joined UCLA in 2005 from New York University, where he held the Lucy and Henry Moses Endowed Professorship in Science and was Professor of Computer Science and Mathematics at NYU's Courant Institute of Mathematical Sciences. Previously, he was Professor of Computer Science and Professor of Electrical and Computer Engineering at the University of Toronto. Before becoming an academic in 1989, he was a Program Leader at Schlumberger corporate research centers in California and Texas.