

A Comprehensive Review of Modern Object Segmentation Approaches

Other titles in Foundations and Trends® in Computer Graphics and Vision

Deep Learning for Multimedia Forensics

Irene Amerini, Aris Anagnostopoulos, Luca Maiano and Lorenzo Ricciardi Celsi

ISBN: 978-1-68083-854-1

Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art

Joel Janai, Fatma Güney, Aseem Behl and Andreas Geiger

ISBN: 978-1-68083-688-2

Discrete Graphical Models - An Optimization Perspective

Bogdan Savchynskyy

ISBN: 978-1-68083-638-7

Line Drawings from 3D Models: A Tutorial

Pierre Bénéard and Aaron Hertzmann

ISBN: 978-1-68083-590-8

Publishing and Consuming 3D Content on the Web: A Survey

Marco Potenziani, Marco Callieri, Matteo Dellepiane and Roberto Scopigno

ISBN: 978-1-68083-536-6

Crowdsourcing in Computer Vision

Adriana Kovashka, Olga Russakovsky, Li Fei-Fei and Kristen Grauman

ISBN: 978-1-68083-212-9

A Comprehensive Review of Modern Object Segmentation Approaches

Yuanbo Wang

Invitae Corporation
mcdy143@gmail.com

Unaiza Ahsan

The Home Depot
unaiza_ahsan@homedepot.com

Hanyan Li

Indeed, Inc.
f1annlee@gmail.com

Matthew Hagen

Amazon.com, Inc.
mathage@amazon.com

now

the essence of knowledge

Boston — Delft

Foundations and Trends® in Computer Graphics and Vision

Published, sold and distributed by:

now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:

now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

Y. Wang *et al.*. *A Comprehensive Review of Modern Object Segmentation Approaches*. Foundations and Trends® in Computer Graphics and Vision, vol. 13, no. 2-3, pp. 111–283, 2022.

ISBN: 978-1-63828-071-2

© 2022 Y. Wang *et al.*

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanical, photocopying, recording or otherwise, without prior written permission of the publishers.

Photocopying. In the USA: This journal is registered at the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923. Authorization to photocopy items for internal or personal use, or the internal or personal use of specific clients, is granted by now Publishers Inc for users registered with the Copyright Clearance Center (CCC). The ‘services’ for users can be found on the internet at: www.copyright.com

For those organizations that have been granted a photocopy license, a separate system of payment has been arranged. Authorization does not extend to other kinds of copying, such as that for general distribution, for advertising or promotional purposes, for creating new collective works, or for resale. In the rest of the world: Permission to photocopy must be obtained from the copyright owner. Please apply to now Publishers Inc., PO Box 1024, Hanover, MA 02339, USA; Tel. +1 781 871 0245; www.nowpublishers.com; sales@nowpublishers.com

now Publishers Inc. has an exclusive license to publish this material worldwide. Permission to use this content must be obtained from the copyright license holder. Please apply to now Publishers, PO Box 179, 2600 AD Delft, The Netherlands, www.nowpublishers.com; e-mail: sales@nowpublishers.com

Foundations and Trends® in Computer Graphics and Vision

Volume 13, Issue 2-3, 2022

Editorial Board

Editor-in-Chief

Aaron Hertzmann

Adobe Research, USA

Editors

Marc Alexa
TU Berlin

Kavita Bala
Cornel

Ronen Basri
*Weizmann Institute of
Science*

Peter Belhumeur
Columbia University

Andrew Blake
Microsoft Research

Chris Bregler
Facebook-Oculus

Joachim Buhmann
ETH Zurich

Michael Cohen
Facebook

Brian Curless
University of Washington

Paul Debevec
*USC Institute for Creative
Technologies*

Julie Dorsey
Yale

Fredo Durand
MIT

Olivier Faugeras
INRIA

Rob Fergus
NYU

William T. Freeman
MIT

Mike Gleicher
University of Wisconsin

Richard Hartley
*Australian National
University*

Hugues Hoppe
Microsoft Research

C. Karen Liu
Stanford

David Lowe
*University of British
Columbia*

Jitendra Malik
Berkeley

Steve Marschner
Cornell

Shree Nayar
Columbia

Tomas Pajdla
Czech Technical University

Pietro Perona
*California Institute of
Technology*

Marc Pollefeys
ETH Zurich

Jean Ponce
Ecole Normale Supérieure

Long Quan
HKUST

Cordelia Schmid
INRIA

Steve Seitz
University of Washington

Amnon Shashua
Hebrew University

Peter Shirley
University of Utah

Noah Snavely
Cornell

Stefano Soatto
UCLA

Richard Szeliski
Microsoft Research

Luc Van Gool
KU Leuven and ETH Zurich

Joachim Weickert
Saarland University

Song Chun Zhu
UCLA

Andrew Zisserman
Oxford

Editorial Scope

Topics

Foundations and Trends® in Computer Graphics and Vision publishes survey and tutorial articles in the following topics:

- Rendering
- Shape
- Mesh simplification
- Animation
- Sensors and sensing
- Image restoration and enhancement
- Segmentation and grouping
- Feature detection and selection
- Color processing
- Texture analysis and synthesis
- Illumination and reflectance modeling
- Shape representation
- Tracking
- Calibration
- Structure from motion
- Motion estimation and registration
- Stereo matching and reconstruction
- 3D reconstruction and image-based modeling
- Learning and statistical methods
- Appearance-based matching
- Object and scene recognition
- Face detection and recognition
- Activity and gesture recognition
- Image and video retrieval
- Video analysis and event recognition
- Medical image analysis
- Robot localization and navigation

Information for Librarians

Foundations and Trends® in Computer Graphics and Vision, 2022, Volume 13, 4 issues. ISSN paper version 1572-2740. ISSN online version 1572-2759. Also available as a combined paper and online subscription.

Contents

1	Introduction	3
1.1	Overview	3
1.2	Convolutional Neural Networks	3
1.3	Object Detection	5
1.4	Object Segmentation	7
2	Traditional Methods in Image Segmentation	9
3	Deep Models for Semantic Segmentation	12
3.1	Fully Convolutional Networks	13
3.2	Deconvolution-based Segmentation	14
3.3	Context-based Segmentation	16
3.4	RNN-based Segmentation	36
3.5	GAN-based Segmentation	39
3.6	Meta-learning for Network Structures	41
3.7	Domain Adaptation	43
3.8	Semantic Segmentation in Large Field of View (FoV) Images	44
3.9	Polarization Driven Semantic Segmentation	47
3.10	Real-time Semantic Segmentation	47
3.11	Discussion	49

4	Deep Models for Instance Segmentation	57
4.1	R-CNN-based Methods (Two-stage)	58
4.2	One-stage Methods	63
4.3	Query-based Models	68
4.4	Other Methods	68
4.5	Discussion	71
5	Deep Learning Models for 3D and Video Segmentation	76
5.1	Voxel-based Semantic Segmentation	77
5.2	Point Cloud-based Semantic Segmentation	78
5.3	RGB-D-based Semantic Segmentation	83
5.4	Other Models for 3D Data Segmentation	84
5.5	Video Object Segmentation	86
5.6	Video Semantic Segmentation	87
5.7	Discussion	89
6	Deep Learning Models for Panoptic Segmentation	91
6.1	Top-down Methods (Two-stage)	93
6.2	Top-down Methods (One-stage)	95
6.3	Bottom-up Methods	95
6.4	Single-path Methods	96
6.5	Weakly-supervised Learning	97
6.6	Video Panoptic Segmentation	97
6.7	Panoptic Segmentation on LiDAR Point Clouds	98
6.8	Discussion	98
7	Datasets	103
8	Evaluation Metrics	113
8.1	Pixel Accuracy / Accuracy	113
8.2	Mean Intersection Over Union (mIoU)	113
8.3	Mean Average Precision (mAP)	114
8.4	Mean Average Recall (mAR)	114
8.5	Dice Score (Coefficient)	115
8.6	Panoptic Quality (PQ)	115
8.7	Video Panoptic Quality (VPQ)	116

9 Challenges and Future Directions	117
9.1 Challenges	117
9.2 Future Directions	120
10 Conclusion	123
Acknowledgements	125
References	126

A Comprehensive Review of Modern Object Segmentation Approaches

Yuanbo Wang¹, Unaiza Ahsan², Hanyan Li³ and Matthew Hagen⁴

¹*Invitae Corporation, USA; mcdy143@gmail.com*

²*The Home Depot, USA; unaiza_ahsan@homedepot.com*

³*Indeed Inc., USA; f1annlee@gmail.com*

⁴*Amazon.com, Inc., USA; mathage@amazon.com*

ABSTRACT

Image segmentation is the task of associating pixels in an image with their respective object class labels. It has a wide range of applications in many industries including healthcare, transportation, robotics, fashion, home improvement, and tourism. Many deep learning-based approaches have been developed for image-level object recognition and pixel-level scene understanding — with the latter requiring a much denser annotation of scenes with a large set of objects. Extensions of image segmentation tasks include 3D and video segmentation, where units of voxels, point clouds, and video frames are classified into different objects. We use “Object Segmentation” to refer to the union of these segmentation tasks. In this monograph, we investigate both traditional and modern object segmentation approaches, comparing their strengths, weaknesses, and utilities. We examine in detail the wide range of deep learning-based segmentation techniques developed in recent years, provide a review of

Yuanbo Wang, Unaiza Ahsan, Hanyan Li and Matthew Hagen (2022), “A Comprehensive Review of Modern Object Segmentation Approaches”, *Foundations and Trends® in Computer Graphics and Vision*: Vol. 13, No. 2-3, pp 111–283. DOI: 10.1561/06000000097.

©2022 Y. Wang *et al.*

the widely used datasets and evaluation metrics, and discuss potential future research directions.

1

Introduction

1.1 Overview

Automated visual recognition tasks such as image classification, image captioning, object detection and image segmentation are essential for image and video processing. Before the advent of deep neural networks, traditional techniques leveraged hand-crafted heuristics to extract visual features and manually tuned parameters to combine these features for inferences and decisions. These techniques are simple yet effective for many cases. However, they are often not generalizable and difficult to configure. For example, Canny Edge Detection, an algorithm that uses Gaussian filters and thresholding to identify edges in images have two key adjustable parameters-size of filters and edge strength thresholds that need to be tuned. If not selected carefully, these parameters can greatly impact the effectiveness of the algorithm, resulting in either missing or false positive edges.

1.2 Convolutional Neural Networks

With the advent of the deep learning era, more powerful and accurate automated visual recognition techniques were enabled by deep neural

networks that can handle high-dimensional data. In particular, convolutional neural networks (CNN) (LeCun *et al.*, 1999) excel at image recognition and pattern identification due to its ability to capture space invariance of shapes in images. CNNs achieve this through two major components: the convolutional layer, pooling layer, and non-linear layers. The convolutional layers apply sliding kernels to extract features from the input image, starting from low-level features such as edges in the earlier layers, to high-level features such as eyes and nose in the later layers. The pooling layer then reduces the resolution of the output feature maps from the convolutional layers, allowing the network to achieve translational and deformation invariance. LeNet-5, developed by LeCun *et al.* (1998), is the first convolutional neural network applied to the famous handwritten digit recognition task MNIST.¹ Although only consisting of a pair of consecutive convolutional and pooling layers followed by three fully connected layers, LeNet-5 is able to achieve a 98.49% accuracy on the MNIST dataset. However, when applied to The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Deng *et al.*, 2009) task, the test accuracy drops to only 66%. Compared with the human level accuracy of 94%, LeNet-5 is not sufficient for more challenging visual recognition tasks.

In 2012, a breakthrough was made when Hinton's team won the ILSVRC challenge with deep learning (Krizhevsky *et al.*, 2012). The challenge included the largest image dataset of the time, totaling over one million images spanning 1,000 object categories. For the first three years since the challenge commenced, most visual recognition systems did not make any breakthrough at this image classification task. The deep learning model proposed by Krizhevsky *et al.* (2012) was named AlexNet, which is often considered the first deep convolutional network. It is similar in structure with LeNet-5, but consists of 3 more convolutional layers and has a total of 62 million trainable variables. It significantly boosted the performance of previous state-of-art image classification techniques by achieving considerably better classification accuracy in the ILSVRC challenge than the second place, reducing the top-5 error rate by over 10% (Krizhevsky *et al.*, 2012). It was not until then that

¹<http://yann.lecun.com/exdb/mnist/>

deep learning started to take over as the go-to approach for challenging computer vision tasks.

Despite the significant improvement AlexNet made towards image classification, it uses too many parameters and is therefore intractable for large-scale training. VGGNet (Simonyan and Zisserman, 2014) solves this problem by replacing the large kernel-sized convolutional layers with multiple layers containing 3×3 kernels. The network contains a total of 13 convolutional layers and 3 fully connected layers. The authors demonstrate that many small-sized filters achieve the same functions as fewer filters with large kernel sizes, but with much fewer parameters.

As CNNs continue to develop and grow deeper, a problem that often arises during the training stage is vanishing gradient – weights disappearing during back-propagation, which hinders the performance of models. ResNet (He *et al.*, 2016), developed one year after VGGNet (Simonyan and Zisserman, 2014), leverages two types of skip connections – identity and projection, to tackle this issue. In addition, it also uses batch normalization (Ioffe and Szegedy, 2015) to stabilize and increase training speed.

Developed by Google, The Inception (Szegedy *et al.*, 2015) network is aimed to capture sparse salient features of varying sizes. It achieves this through stacking multiple sizes (5×5 , 3×3 , and 1×1) of kernels at the same layer and combining outputs to feed to the next layer. 1×1 convolutions are used to reduce the number of channels, thereby reducing the number of parameters of the network.

1.3 Object Detection

A precursor to the image segmentation problem, object detection not only classifies, but also localizes each object in an image. One of the earlier approaches to solving object detection is the sliding window approach (Bosch *et al.*, 2007; Chum and Zisserman, 2007; Dalal and Triggs, 2005; Ferrari *et al.*, 2007; Rowley *et al.*, 1995). By taking different crops from an input image and applying CNNs to classify cropped regions (a cropped region is classified as “background” if no target object is identified), different objects are simultaneously localized and categorized from the input image.

However, because objects appear in various sizes and in many locations in the image, it is very computationally expensive to apply CNNs to a potentially infinite set of cropped input regions. Instead, a more scalable approach called region proposals (Alexe *et al.*, 2012; Uijlings *et al.*, 2013; Zitnick and Dollár, 2014) was developed to find candidate regions in images that are likely to contain objects quickly based on traditional image and signal processing techniques. One example of such a technique is Selective Search, which uses sub-segmentation and greedy search to generate and recursively combine regions into a final list of 2,000 candidate regions (Uijlings *et al.*, 2013). With region proposals approaches, CNNs can then be applied to a much smaller set of cropped regions to produce final detection outputs, thus making this task much more computationally tractable.

The idea of combining region proposals and CNNs for object detection first materialized in Girshick *et al.* (2014). After the features are extracted from the proposed regions using CNNs, an SVM is used to 1) determine whether an object is present in the proposed region and classifies it if present, and 2) perform bounding box regression to compute offsets to the proposed region and refine its boundary. Although R-CNN is able to resolve the challenges with the sliding window approach to object detection, it is still very slow since CNNs need to be applied to 2,000 regions, and selective search could generate bad proposals since it is a fixed algorithm.

The same authors of R-CNN developed Fast R-CNN (Girshick, 2015) to address the slow selective search stage. Instead of applying CNNs to proposed regions, the input image is fed into the CNNs directly to generate a feature map. From the feature map, candidate regions are produced and an RoI pooling layer is then applied to reshape these regions into fixed sizes before they are fed into a fully connected region to produce a feature vector. Finally, a softmax layer is applied to classify this feature vector into objects, and a bounding box regressor is used to compute the offset values to refine boundaries. This is a much faster approach than R-CNN because the CNNs need only be run one time on the input image instead of on 2,000 regions. Although Fast-R-CNN is significantly faster than R-CNN during both training and testing, the region proposal stage remains to be the bottleneck. This is because it

still uses selective search on the CNN feature map to propose candidate regions.

Faster R-CNN (Ren *et al.*, 2015) is a similar approach to both algorithms above, but leverages a separate network on the CNN feature map to propose candidate regions instead of using selective search. As a result, Faster R-CNN beats its predecessors in inference speed and can generate detection results in real-time.

Whereas the object detection models from the R-CNN family use regions to identify objects in an image, YOLO (Redmon *et al.*, 2016) is an approach that looks at not just the regions of interest, but the entire image. The image is divided into grids. A set of base boxes centered at each grid cell is fed as input to a single CNN to predict both object class scores and bounding boxes. This approach is extremely fast as detection is reframed as a regression problem. YOLO also sees the entire image when making predictions and makes fewer mistakes compared to Fast R-CNN when identifying background patches (Redmon *et al.*, 2016). However, it is not good at predicting small objects due to the constraints placed on the base boxes.

Similar to YOLO, SSD (Liu *et al.*, 2016a) is a single-shot (achieves object localization and detection simultaneously) object detector with fast inference speed to enable real-time detection. It uses a modified version of MultiBox (Erhan *et al.*, 2014), a bounding box regression technique, to generate candidate bounding boxes on each grid cell of the feature map. Given a set of default bounding boxes with different aspect ratios, SSD selects those with an intersection over union (IOU) score larger than 0.5 with the ground truth bounding box as candidates and performs bounding box regression to refine the boundaries. It also predicts, for each object class c and each candidate bounding box b , the probabilities of box b containing object c .

1.4 Object Segmentation

While object detection aims only to localize objects with bounding boxes and classifying localized objects, object segmentation adds the task of delineating precise object boundaries by classifying every unit in an input to an object class or the background. The general formulation for this

problem can be defined as follows: given an input space $\mathcal{X}$, and a label space $\mathcal{L}$, object segmentation finds a mapping $f: \mathcal{X} \mapsto \mathcal{L}$. It has a wide range of applications in many industries including healthcare (e.g. cancer detection), transportation (self-driving), robotics (guidance), fashion (virtual try-ons), home improvement (decor visualizers), and tourism (destination recognition). Segmentation of an image breaks it down into simplified representations that can aid downstream interpretation tasks. Thus, segmentation differs from detection as it does not have to be tied to a specific task, but rather an intermediate step in machine perception. Nevertheless, it is more challenging and usually more time-consuming than object detection, thereby requiring more advanced techniques and more high-quality annotated training data. Object segmentation is fundamentally tied to human image perception as recognition of objects from an input image can be conceptualized into a process of dissecting regions of deep concavity into simple volumetric components such as blocks, wedges, and cones (Biederman, 1985).

Recognition-by-components (RBC) theory states that robust object perception is possible because detection of edge properties such as curvature, symmetry, and parallelism is usually invariant over image quality and viewing angle (Biederman, 1985). However, because machines process images through various means unlike the process of human interpretation, it is far from enough to only focus on these components when performing object segmentation. In this monograph, we review various techniques developed over the years for the task of object segmentation, with a focus on the deep learning-based techniques for the two most widely solved segmentation tasks: Semantic Segmentation and Instance Segmentation. We also survey methods developed for Panoptic, Video and 3D data segmentation. We categorize and compare various techniques developed for each task. The readers will find that various themes emerge from these techniques that push machines to their limits and often times deviate from human perception principles. In addition, we provide an overview of the widely used benchmark datasets for each of these techniques, along with the respective evaluation metrics to measure the models' performances. Finally, we discuss potential directions for future research in these areas.

2

Traditional Methods in Image Segmentation

Many traditional methods have been proposed for image segmentation. Otsu's method uses the threshold between classes with the aim to minimize the intra-class variance (Otsu, 1979). Seeded region growing, a region-based approach, uses initial points to expand or grow the regions and then makes up the segmented objects (Adams and Bischof, 1994; Qin *et al.*, 2010; Ramli and Tahar, 2020). The Edge detection method identifies pixels representing object edges in the images, which are then used to determine the labels of the remaining pixels (Rong *et al.*, 2014; Canny, 1986). Level set techniques based on Hamilton-Jacobi formulations (Osher and Sethian, 1988) are also widely used (Malladi *et al.*, 1995; Luo *et al.*, 2019; Yan *et al.*, 2020a; Yang *et al.*, 2017). Graph-based algorithms use a data structure consisting of vertices and weighted edges, and some typical methods are Normalized cuts (Shi and Malik, 2000), Graph cuts (Moore *et al.*, 2008), and Entropy Rate Superpixel Segmentation (Liu *et al.*, 2011b) algorithms. The mean shift method is based on density estimations in a feature space and efficiently finds peaks in a high-dimensional data distribution without explicitly computing the complete function (Comaniciu and Shift, 2002; Szeliski, 2010). Mumford-Shah functional finds the edge set and a piecewise

smooth approximation given the image intensity function (Mumford and Shah, 1989).

Active Contour Models (ACMs) is another traditional gradient-based method first introduced by Kass *et al.* (1988). It can be applied to various tasks such as edge detection, motion tracking, and segmentation. It formulates segmentation as an energy minimization problem and uses partial differential equations (PDEs) – based methods for optimizing active contours. Several variations of ACMs have also been developed including active contour without edge (ACWE) (Chan and Vese, 2001) and fast global minimization-based active contour model (FGM-aCM) (Chen *et al.*, 2019c). Recent works have combined ACMs and deep neural networks for image segmentation.

Chen *et al.* (2019c) propose an active contour – based loss function that combines region and contour length information of segmentation masks for CVD diagnosis. The base architecture used for segmentation is Dense block based U-Net (Dense-Net) (Huang *et al.*, 2017). The proposed loss function has superior performance compared with the common Cross-entropy loss when using either U-Net (Ronneberger *et al.*, 2015) or Dense-Net (Huang *et al.*, 2017).

Gur *et al.* (2019) develop another unsupervised loss function based on morphological ACWE (Chan and Vese, 2001). A novel pooling layer is developed that mimics curvature morphological operators, examples of which include dilation and erosion operations. The main network is an encoder-decoder based architecture with successive morphological pooling layers for smoothing. It achieves the state-of-the-art performance on the DeepVess (Haft-Javaherian *et al.*, 2019) and VesselNN (Teikari *et al.*, 2016) datasets .

Besides incorporating ACMs into loss functions, ACMs have also been used as a means for post-processing outputs from FCNs (Le *et al.*, 2018; Rupprecht *et al.*, 2016). Trainable Deep Active Contours (TDACs) (Hatamizadeh *et al.*, 2020) is one of the most recent works that fully integrates ACM and CNNs into an end-to-end trainable architecture. The backbone CNN directly predicts per-pixel 2D parameter maps to initiate and control a Eulerian energy functional that the ACM utilizes. Each term of the energy functional is automatically differentiable with a TensorFlow implementation framework, allowing for optimiza-

tion through stochastic gradient descent. The Eulerian ACM energy model improves the boundaries of the object segmentation by adjusting the contours. This work achieves state-of-the-art performance on the Vaihingen buildings dataset (Gerke, 2014) and the Bing Huts dataset (Marcos *et al.*, 2018). Other works integrating ACM into deep networks for image segmentation include Hu *et al.* (2017), Hatamizadeh *et al.* (2019a, 2019b), Cheng *et al.* (2019), Wang *et al.* (2019e), Hoogi *et al.* (2016), Akbarimoghaddam *et al.* (2022), and Peng *et al.* (2020).

Traditional approaches often focus on extracting the lower level features such as edges, shapes, and gradients from the scene. These methods have low demands for computational resources, but are limited in performance. They are often challenged by complex scenes or obscure objects, and sometimes require manual parameter tuning. Nevertheless, traditional methods provide the foundation for many advanced deep learning models for object segmentation, and many modern approaches have continued to leverage certain components from these methods to boost their performances.

3

Deep Models for Semantic Segmentation

Semantic segmentation is a form of dense prediction where each pixel from an image is associated with a class label. If multiple instances of the same object class exist in an image, there is no need to distinguish each individual instance with a different label. For clarity, this section only considers Semantic image segmentation, which takes as input either a colored RGB image with three channels ($height \times width \times 3$) or a grayscale image ($height \times width \times 1$) and produces a label map ($height \times width \times 1$), with each pixel (index) assigned an integer that represents an object (class) label corresponding to the predicted object type at the corresponding pixel location in the input image. Mathematically, semantic image segmentation can be formulated as the following: given an input image $\mathcal{P} = p_1, p_2, \dots, p_m$, and a label space $\mathcal{L} = l_1, l_2, \dots, l_n$, it assigns an element from $\mathcal{L}$ to each element of $\mathcal{P}$.

Segmentation by itself can be defined without a specific task. Prior to the deep learning era, due to the limited capacity of traditional computer vision methods, segmentation was limited to “processing” of images, where the output included regions and structures such as edges (lines and curves) and shapes, which can be obtained with low-level image information such as gradients and colors (Arbelaez *et al.*, 2010). As deep

learning technologies matured, semantic segmentation has evolved from “processing” to “recognizing and understanding” the images, particularly on a dense, pixel-level. Semantic segmentation plays an important role in the medical domain, autonomous driving, and modern robot vision and understanding. This section reviews various methods that focus on solving the problem of semantic segmentation.

3.1 Fully Convolutional Networks

One of the earliest influential methods towards semantic segmentation is the Fully Convolutional Networks (FCN) (Long *et al.*, 2015). By removing the last fully connected layer from traditional CNN networks, FCN generates a heatmap that preserves the spatial information from the input image. This feature map is then up-sampled using fixed interpolation techniques such as bilinear interpolation or using learned deconvolution layers. In addition, skip-connections are added between final up-sampling layers and the earlier feature maps so that visual information at different granularities can be combined to produce the final segmentation output. The entire architecture is shown in Figure 3.1. FCN achieves 62.7% mean IOU on the PASCAL VOC 2011 (Everingham *et al.*, 2010) dataset, approximately 10% better than the previous state-of-art using SDS (Hariharan *et al.*, 2014). It also reduces inference time by 286x overall. However, FCN remains to be limited by its high computational requirements due to the number of parameters and its slow real-time inference speed. More importantly, by using only convolution layers, FCN only captures local information when making pixel inferences. Additionally, it suffers from rough segmentation boundaries due to the loss of information during the up-sampling stages.

The portion of the FCN responsible for down-sampling and feature map generation is an encoder, which is usually a base classification model such as VGGNet or ResNet stripped of its fully connected layers. The deconvolution / up-sampling portion of the FCN is a decoder, which produces pixel-wise predictions based on the feature map output from the encoder. Many semantic segmentation approaches are based on this encoder-decoder structure, with variations mainly at the decoder portion of the network.

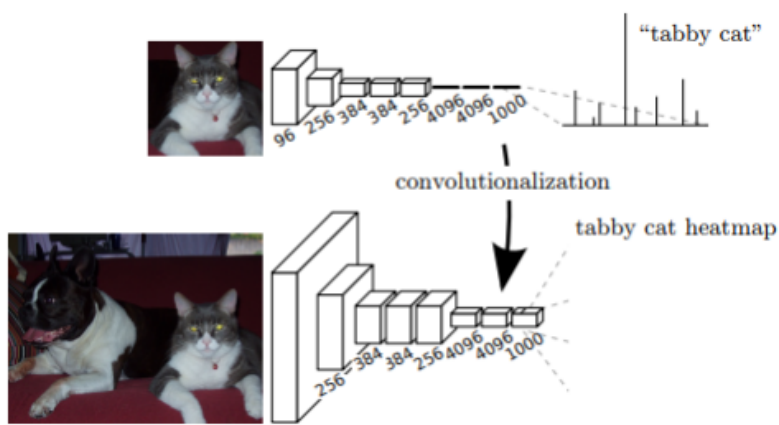


Figure 3.1: Architecture of the Fully Convolutional Networks. From Long *et al.* (2015).

U-Net

The U-Net (Ronneberger *et al.*, 2015) is based on FCN, resembling its encoder-decoder structure. U-Net addresses the issue of FCN’s information loss by adding shortcut connections between each pair of corresponding down-sampling and up-sampling layers so that the earlier layers that capture lower-level details of the input image can propagate such information to the up-sampling layers to help with more precise localization. U-Net was developed and widely used for biomedical imaging tasks and has been later extended into various modified architectures (Quan *et al.*, 2016; Li *et al.*, 2018f; Shah *et al.*, 2018) for other tasks.

3.2 Deconvolution-based Segmentation

Rather than using direct upsampling in the decoder portion of the network to recover the original dimension of the input image for the final dense prediction map, deconvolution-based segmentation approaches use deconvolution layers to learn how to expand the output feature map from the encoder layers. This allows the model to learn non-linear upsampling with more model parameters, whereas bilinear upsampling used in previous approaches only allow linear upsampling of encoded

features with fewer model parameters but a faster computation speed. Based on the FCN architecture, DeconvNet (Noh *et al.*, 2015) is the first method that applies trainable filters in the decoder portion of the network to generate dense feature maps.

SegNet

The SegNet (Badrinarayanan *et al.*, 2017) model also uses the encoder-decoder structure and is a lightweight version of DeconvNet (Noh *et al.*, 2015). Instead of transferring the entire feature maps from down-sampling to up-sampling layers like U-Net, it uses pooling indices computed from the max pooling layers during the down-sampling stage when performing deconvolution so that the up-sampling process is non-linear. This reduces the memory required and consequently model size. Bayesian SegNet (Kendall *et al.*, 2017) is a modified version of SegNet that aims to add a probability measure to the segmentation output, and demonstrates that by modeling this uncertainty, state-of-art model architectures benefit from improved segmentation performance without additional parameters.

SDN

Stacked Deconvolutional Network (SDN) (Fu *et al.*, 2019b) aims to improve the resolution of the segmentation results progressively through stacking of SDN units. Each SDN unit is a shallow deconvolutional network composed of two down-sampling blocks followed by two up-sampling blocks. Intra-unit connections links inputs between two consecutive convolutional layers within each SDN unit. Inter-unit connections are similar to the skip connections employed in FCN that links corresponding down-sampling and up-sampling blocks. These dense connections allow efficient reuse of multiscale information across SDN units but increase GPU memory usage. Compression units that reduce feature map channels are therefore employed to reduce the computational demands. Figure 3.2 shows the architecture of the SDN. SDN achieves a superior performance of 86.6% MIoU on the PASCAL VOC 2012 (Everingham *et al.*, 2010) dataset, making it one of the best among the earlier semantic segmentation methods.

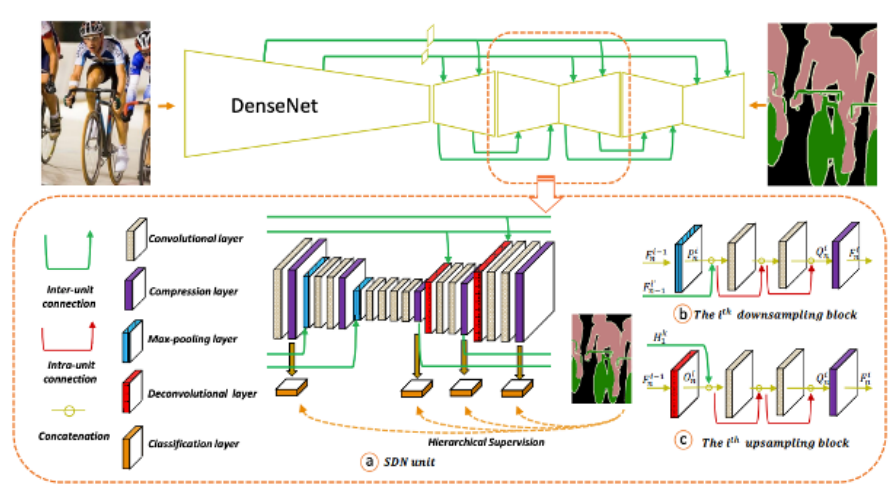


Figure 3.2: Overall SDN architecture (upper) and SDN unit (lower). From Fu *et al.* (2019b).

LSD-Net

The Locality-Sensitive Deconvolution Networks (LSD-Net) (Cheng *et al.*, 2017b) is developed to improve two aspects of the deconvolution layers: boundary refinement and RGB-D fusion. It achieves the former by incorporating visual cues from raw RGB-D data, and the latter through a gated fusion layer that learns to adjust the RGB and depth's contributions at each pixel location. Through these enhancements, the model is able to achieve state-of-art performance on the NYU-Depth v2 (Silberman *et al.*, 2012) dataset.

3.3 Context-based Segmentation

Pixel classification can be a difficult process due to the confusion made when the network only relies on local information – a drawback of FCN-based architectures. These models address this issue by incorporating contextual knowledge when predicting the semantic label for a pixel. Various forms of image contexts can be encoded using different approaches, including multi-resolution feature maps, atrous (dilated) convolutions, recurrent networks, attention, and Conditional Random Fields.

3.3.1 Multi-resolution Features-based Segmentation

By taking into account multi-resolution features, these models often achieve better performances at the expense of increased model parameters and slower computational speeds. Many approaches have successfully captured and fused multiscale features for dense pixel predictions and will be discussed in this section.

FPN

Feature Pyramids (Figure 3.3) is a common approach for encoding multiscale features. The Feature Pyramid Network (FPN) model (Lin *et al.*, 2017b) is originally developed for object detection task and consists of a top-down and a bottom-up pathway. The bottom-up pathway computes hierarchical feature maps in a feedforward fashion using a backbone network. The top-down pathway up-samples feature maps from higher pyramid levels using nearest neighbor, while lateral connections between the same spatial levels of the two pathways are merged with element-wise addition to produce enhanced features. After applying a 3x3 convolution to each of the final feature maps to reduce up-sampling's aliasing effect, a prediction is generated from each level for object detection. For image-segmentation tasks, a multi-layer perceptron (MLP) is applied to each level of the feature pyramid to predict the segmentation masks and object scores (Lin *et al.*, 2017b).

PSPNet

The Pyramid scene parsing network (PSPNet) (Zhao *et al.*, 2017) features a CNN backbone, dilated convolutions, and a Pyramid Pooling Module which allows the network to take global information into account while making local predictions. The feature map output from the encoder layers are pooled at 4 different resolutions and 1x1 convolutions are applied to reduce the channel dimensions. These outputs are standardized through up-sampling and fused together, and finally merged with the original feature map. Final segmentation predictions are generated after a convolutional layer is applied to the final concatenated feature map. The Pyramid Pooling Module has laid the foundation for

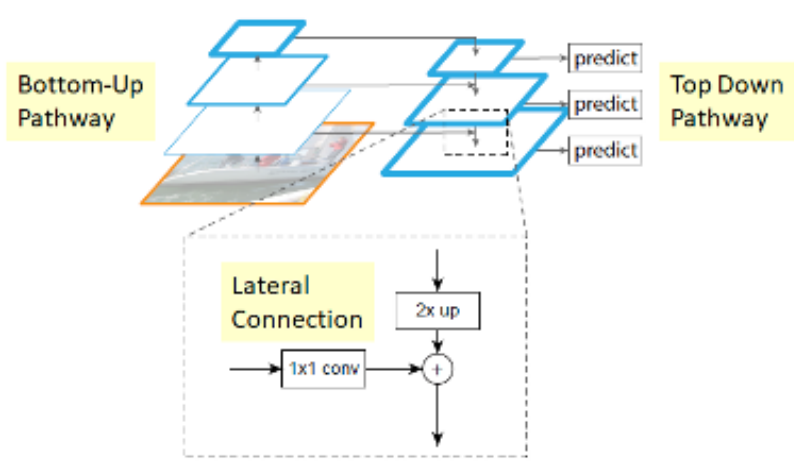


Figure 3.3: Feature Pyramids. From Lin *et al.* (2017b).

many later works including the Deeplab (Chen *et al.*, 2017a; 2017b) family of approaches.

DMNet

The Dynamic multiscale Network (DMNet) (He *et al.*, 2019a) is another model that leverages feature maps at multiple resolutions to generate segmentation predictions. The Dynamic Convolutional Module uses context-aware filters whose kernel sizes are dynamically determined based on the input. DMNet applies multiple Dynamic Convolutional Modules (DCMs) to the feature map output from the encoder layers in parallel to generate multiscale feature maps, which are then fused with the original feature map and passed into a convolutional layer to produce final pixel-level predictions.

CCN

Ding *et al.* (2018) developed a context contrasted local model that simulates how human processes contextual information when identifying an object. More specifically, it introduces a function that allows the network to generate discriminative context features for an object and is thus able to disregard more dominant but irrelevant features when

classifying the target pixel. This framework generates multi-level context-aware local features and uses gated sum to aggregate features selectively before making final predictions.

MSCI

Lin *et al.* (2018) introduce a novel method called Multiscale Context Intertwining (MSCI) for combining local and global contextual information to improve semantic segmentation. It allows pairs of high- and low-resolution feature maps to exchange mutual contextual information through bidirectional connections inspired by long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997). Specifically, it allows context information propagation between different regions of the input image adaptively by computing super-pixels based on image structures and establishing connection pathways based on their relationships. LSTM units are trained to propagate information along these pathways to build more powerful feature maps that combine multi-feature scales.

ParseNet

ParseNet (Liu *et al.*, 2016b) uses a simple technique to incorporate global context into semantic label predictions. Specifically, a ParseNet module first applies global average pooling to a feature map to obtain global features. L2 normalization is then performed, and an unpooling layer simply replicates the values in the global feature vector so that the output is the same size and can be finally combined with the original feature map. The combined feature map is then used for final predictions.

RefineNet

The multi-path refinement network (RefineNet) (Lin *et al.*, 2017a) is designed to address the loss of image resolution during the downsampling process in the semantic segmentation networks. RefineNet employs long-range residual connections (He *et al.*, 2016) to enable low-level information from the downsampling process to transfer to and refine the high-level feature maps. Local short-range residual connections are

incorporated in residual Convolution Units (RCUs) and residual pooling components. These residual connections allow direct back propagation of gradients so that the network can be trained from end-to-end. In addition, this paper also designs chained residual pooling to capture contextual information from a large image region. It uses an efficient structure to fuse multiscale pooled features. Specifically, multiple pooling blocks are connected in chains, allowing later blocks to reuse earlier ones' outputs so that features from larger regions is accessible without a large pooling window. RefineNet achieved state-of-the-art performance on seven public datasets.

FaPN

In FPN (Lin *et al.*, 2017b), feature maps from the top-down and bottom-up pathways are merged with element-wise addition. While it is effective in capturing long-range information and contextual information, its simplicity could result in the issue of feature misalignment. Huang *et al.* (2021c) design a Feature-aligned Pyramid Network (FaPN) to address this issue. There are two key components of the FaPN framework: a Feature Selection Module (FSM) and a Feature Alignment Module (FAM). The FSM applies a feature importance modeling layer to the global information extracted from each input feature map with global average pooling and learns an importance vector that is used to scale the input feature maps (Lin *et al.*, 2017b). A feature selection layer then selectively drops useless feature maps, leaving important ones behind. FSM replaces the conventional 1x1 convolution for channel reduction. The FAM uses deformable convolution (Dai *et al.*, 2017b) to learn the transformation offset values that guide alignment of up-sampled higher-level features. When integrated in strong baselines, FaPN improved state-of-the-art performance on Cityscapes (Cordts *et al.*, 2016) and COCO-Stuff (Caesar *et al.*, 2018) datasets.

ExFuse

ExFuse (Zhang *et al.*, 2018b) is a feature fusion framework aimed to improve semantic segmentation. The authors argue that simply merging low-level and high-level features is ineffective due to semantic and

resolution gaps (Zhang *et al.*, 2018b). Instead, they attach semantic information to low-level features through layer rearrangement, semantic supervision and semantic embedding branch, and embed spatial information into high-resolution features through explicit channel resolution embedding and densely adjacent prediction (Zhang *et al.*, 2018b). Either approach improves segmentation performance significantly. Using ResNeXt (Xie *et al.*, 2017) as the backbone CNN, ExFuse achieved state-of-the-art performance on the PASCAL VOC 2012 (Everingham *et al.*, 2010) dataset.

APCNet

The Adaptive Pyramid Context Network (APCNet) (He *et al.*, 2019b) replaces the DCMs in the DMNet (He *et al.*, 2019a) with Adaptive Context Modules (ACMs). Similar to DCM, each ACM also has two separate branches. One branch is responsible for estimating Global-guided Local Affinity (GLA) coefficients which assigns weights to different areas based on their contributions to determine the semantic label of local pixels. The weight assignment takes into account both local and global representations. The other branch is used to obtain subregion representations. Outputs from the two branches are multiplied to produce the adaptive context vectors. multiscale context vectors are combined with the feature map output from the encoder before being passed into the final convolutional layer to obtain semantic labels. APCNet achieves the integration of both multiscale and global contextual information.

3.3.2 Atrous-convolution-based Segmentation

Another line of approach for incorporating multiscale contextual information to feature maps is through dilated convolutions (a.k.a. atrous convolutions) (Chen *et al.*, 2017a). This type of convolution inserts gaps into kernels, expanding its field-of-view to incorporate context from a larger area without needing additional parameters. By using atrous convolution kernels of varying gap sizes, models can effectively learn the trade-offs between assimilating more contexts with larger field-of-views and localizing details with smaller field-of-views.

Atrous convolution allows us to enlarge the field of view of filters to incorporate larger context. It thus offers an efficient mechanism to control the field-of-view and finds the best trade-off between accurate localization (small field-of-view) and context assimilation (large field-of-view).

DilatedNet

Yu and Koltun (2015) is one of the first works that proposes a semantic segmentation approach based on aggregation of contextual information through atrous convolutions with multiple sampling rates. Specifically, DilatedNet uses five different rates of dilation, which allows the extraction of multiscale features without losing feature resolution.

DeepLab

DeepLab (Chen *et al.*, 2017a) is a family of models developed at Google that proposed several key techniques to improve quality of segmentation output – atrous convolution, atrous spatial pyramid pooling (ASPP), and the incorporation of conditional random fields (CRFs). By varying the gap size, or sampling rate, atrous convolution kernels extract varying scales of information from the input feature layer. ASPP uses multiple atrous convolution kernels with different sampling rates to capture the object along with multiple scales of contextual information (Figure 3.4).

DeepLab also introduces a Graphical CRFs to address the coarse boundaries of segmentation output resulting from the usage of pooling layers. When classifying a pixel, the fully connected CRF considers other pixel labels. This refines and sharpens boundaries of segmentation results.

DeepLabV3+ (Chen *et al.*, 2018c) adds a decoder module to further refine segmentation output, particularly on boundary refinement. Instead of using a naïve decoder such as the one used in Chen *et al.* (2017b) to simply up-sample features using bilinear interpolation, it first up-samples encoder features by a factor of 4, and then concatenates the partially up-sampled features with corresponding features generated from the backbone CNN with the same spatial resolution. These features

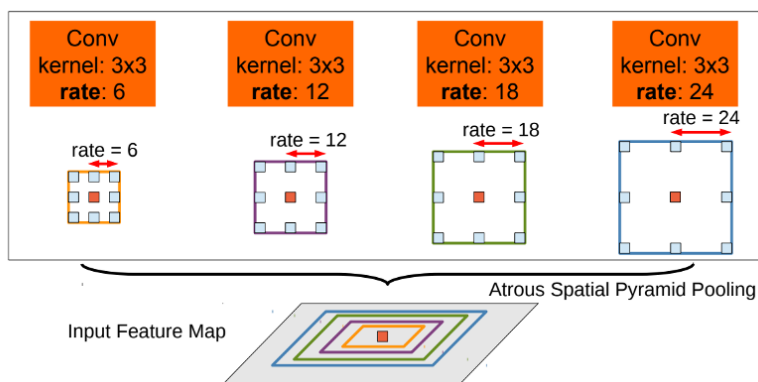


Figure 3.4: Atrous Spatial Pyramid Pooling. From Chen *et al.* (2017a).

are refined through a few 3x3 convolutions before being up-sampled by another factor of 4 to recover the original resolution and generate the final segmentation output.

ResNet-DUC-HDC

Wang *et al.* (2018) identify the “gridding” issue from dilated convolutions and propose Hybrid Dilated Convolutions (HDC) as the solution. In a sequence of dilated convolutions, the sampling rate is gradually increased to improve coverage of the receptive field’s areas, resulting in more accurate pixel-label predictions. In contrast to DeepLab’s ASPP, HDCs are applied sequentially as opposed to in parallel. In addition, the authors also design Dense Upsampling Convolution (DUC) to achieve improved pixel-level decoding. Specifically, instead of using bilinear up-sampling or deconvolution to transform the output feature map from the ResNet encoder before making final predictions, DUC uses a simple yet more effective approach. First output feature map’s dimension of $h \times w \times c$ is used to obtain $h \times w \times (d^2 \times L)$, where L is the number of categories in the segmentation task. Then, the feature map is transformed into $h \times w \times L$ using softmax before element-wise argmax is used to obtain the prediction labels. The key idea is DUC is the division of the whole label map into d^2 parts with the same height and width as the input

feature map. DUC is also learnable by nature and can be trained to capture fine-grained details for semantic label predictions.

DenseASPP

Densely connected Atrous Spatial Pyramid Pooling (DenseASPP) (Yang *et al.*, 2018) was developed to address the issue of sparsity in ASPP, particularly when dealing with autonomous driving scenario, where objects exhibit large variations of scales. DenseASPP organizes atrous convolutions in a densely connected, cascade fashion, so that sampling rate continues to increase through each layer and earlier layers' outputs are concatenated with each later layers' outputs before the next layer of atrous convolution is applied. Without increasing the number of atrous layers, this structure can produce a much denser multiscale feature map.

GALD

The use of global aggregation techniques such as average pooling and spatial-wise feature propagation can help FCNs model long-range dependencies (Li *et al.*, 2019d). However, large patterns often dominate Global Aggregation (GA), which leads to oversmoothed smaller patterns such as boundaries and small objects. Global Aggregation and then Local Distribution (GALD) is a framework for solving this issue. Specifically, from the output feature map of the backbone FCN, the GA module first exploits the long-range contextual information by calculating global statistics for each group of spatial and channel positions and multiplying back to features in the same group (Li *et al.*, 2019d). The Local Distribution (LD) module then adaptively distributes GA features to each spatial and channel position of the output feature (Li *et al.*, 2019d). When used in semantic segmentation, GALD achieves competitive performance on Cityscapes dataset (Cordts *et al.*, 2016).

MRFM

One of the main challenges associated with semantic segmentation is the diversity of the size of the objects in an image. The Multi-Receptive Field Module (MRFM) (Yuan *et al.*, 2020a) is designed to explicitly

incorporate multiscale features to address this challenge. Unlike previous works that integrate multiscale information by modifying network structures attached to the backbone CNNs, MRFM is a redesign of the backbone network itself. Any basic module (e.g. a bottleneck in ResNet) can be replaced with MRFM, which has two separate paths – one is exactly the same as basic module, and the other follows the same structure as the basic module, but replaces basic convolution with atrous convolution. The output of each path is weighted to adaptively select the more suitable receptive field before fusion. The authors of MRFM also develops an edge-aware loss function (Yuan *et al.*, 2020a) to tackle boundary pixel classification. This loss function penalizes misclassified pixels near or on the edge more than other pixels. The Multi Receptive Field Network that combines these two components achieved state-of-the-art results on Cityscapes (Cordts *et al.*, 2016) and Pascal VOC2012 datasets (Everingham *et al.*, 2010).

3.3.3 Attention-based Models

The self-attention mechanism computes representations of current input location by taking into account its relationship with all other positions of the input. In the case of semantic segmentation, self-attention effectively captures contextual information when trying to predict a target object. Attention blocks are often applied in feature concatenation, allowing the network to incorporate features that are more relevant for the prediction task by assigning more weights to them. By nature, attention mechanisms are designed for sequence-to-sequence problems, which requires reformulating the segmentation task, as was done in SEgmentation TRansformer (**SETR**) (Zheng *et al.*, 2021). In addition, the versatility of the attention mechanism is shown by the models that leverage it to achieve a diverse range of sub-tasks such as capturing intra- and inter-class similarity and improving feature representation for more precise predictions.

DANet

The Dual Attention Network (DANet) (Fu *et al.*, 2019a) utilizes attention modules to adaptively incorporate contextual dependencies into

local features to improve pixel-level predictions. It applies two parallel attention modules to the feature map output from a dilated residual network backbone. The position attention module captures semantic similarities between features at all locations using self-attention mechanism. Specifically, similar features at different positions contribute to mutual improvement at each position regardless of their spatial distances. The channel attention module captures interdependent channel maps in a similar fashion. The outputs from the two parallel attention modules are summed to create improved feature representations that lead to more precise segmentation predictions (Fu *et al.*, 2019a). The architecture of DANet is shown in Figure 3.5.

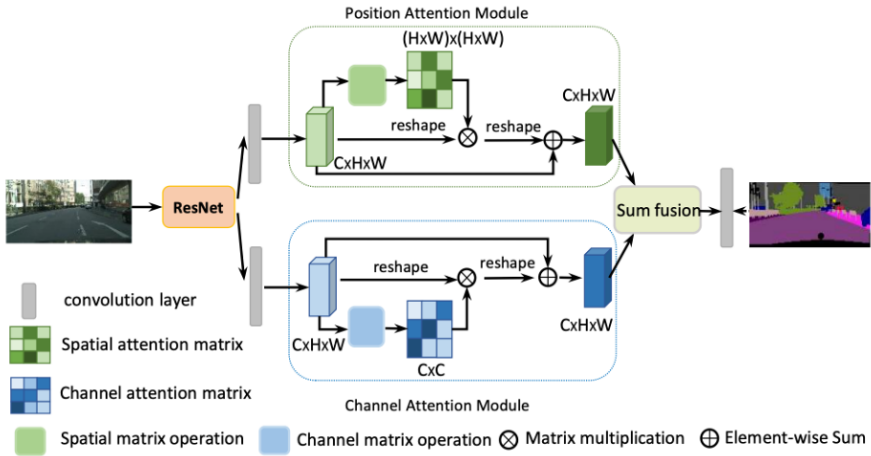


Figure 3.5: Architecture of DANet. From Fu *et al.* (2019a).

PSANet

The Point-wise Spatial Attention Network (PSANet) (Zhao *et al.*, 2018b) is another model that utilizes attention module to involve contextual information to enhance feature maps. The PSA module consists of two parallel branches that generates point-wise spatial attention maps (Zhao *et al.*, 2018b). One branch generates distribute attention (each position distributes information to others), whereas the other branch generates

collect attention (each position collects information from others). The final feature representation is composed of this bidirectional information combined with local features and is used for final segmentation predictions.

GT-OCR

The Object-Contextual Representation (OCR) (Yuan *et al.*, 2020b) is an attention-like model that calculates a set of weights for aggregating object region representations based on the relationships between object regions and contextual pixels. Ground-truth OCR (GT-OCR) (Yuan *et al.*, 2021a) exploits ground-truth labels to improve the estimates of OCR. It is based on the motivation that “the class label assigned to one pixel is the category of the object that the pixel belongs to” (Yuan *et al.*, 2021a). First, supervised by ground-truth segmentation, a backbone CNN computes a coarse soft segmentation that divides the contextual pixels into object regions. Then, each object region’s representation is estimated based on an aggregation of the representations of corresponding pixels. Finally, OCR is used to augment each pixel’s representation. GT-OCR differs from ASPP (Chen *et al.*, 2017a) in that it differentiates between same-object-class and different-object-class contextual pixels. GT-OCR achieves competitive performance on multiple benchmark datasets including Cityscapes (Cordts *et al.*, 2016), ADE20K (Zhou *et al.*, 2017), LIP (Gong *et al.*, 2017), PASCAL-Context (Mottaghi *et al.*, 2014), and COCO-Stuff (Caesar *et al.*, 2018).

SegFormer

SegFormer (Xie *et al.*, 2021) is a simple and lightweight transformer-based network for semantic segmentation. It has two main components: a hierarchical Transformer encoder and an all-MLP decoder. The encoder module is a series of transformers based on ViT (Dosovitskiy *et al.*, 2020) with the same architecture but different sizes (Xie *et al.*, 2021). These transformers produce multiscale features including high-resolution coarse features and low-resolution fine features (Xie *et al.*, 2021), and patch merging is used to generate the final hierarchical feature map. The

large effective receptive field (ERF) realized through this hierarchical transformer encoder allows a lightweight MLP to serve as the decoder, which fuses the multiscale features and predicts a final segmentation mask.

EMA-Net

Resources required to compute self-attention can be demanding. The authors of EMA-Net (Li *et al.*, 2019c) addresses this issue by designing the Expectation-Maximization Attention (EMA) module that estimates a smaller set of bases over image pixels using the EM algorithm. The expectation (AE) step computes the attention map's expectation, and the maximization step (AM) subsequently updates the parameters (bases).

DFN

Authors of the Discriminative Feature Network (Yu *et al.*, 2018c) attempt to improve intra-class consistency and inter-class distinction – two important challenges in semantic segmentation. They design a Channel Attention Block (CAB) based on the attention mechanism to enhance consistency. It introduces a set of weight parameters that is applied to different channels. This effectively selects discriminative features from each stage of the network to improve the consistency of intra-class prediction.

SANet

The squeeze-and-attention network (SANet) (Zhong *et al.*, 2020) incorporates a novel squeeze-and-attention (SA) module to address the implicit task of pixel grouping in semantic segmentation. The authors argue that previous multiscale feature aggregation strategies employed in semantic segmentation models under-exploit the global perspective due to the spatial restrictions enforced by the grid structures of the kernels. The SA module is designed to bypass these restrictions by including an additional path similar to the squeeze-and-excitation (SE) module (Hu *et al.*, 2018a) for re-calibrating output feature maps' channels. Different

from the SE module, SA module does not fully squeeze the feature maps, but instead uses average pooling to downsample them. This mechanism allows attention of pixel groups belonging to the same class at different spatial scales to be emphasized, thus alleviating the local constraints of convolution kernels (Zhong *et al.*, 2020). SANet integrates outputs from multiple SA heads at different resolutions to generate class-wise masks that consider pixel grouping. It achieves state-of-the-art performance on the PASCAL Context dataset (Mottaghi *et al.*, 2014).

Swin Transformer

To address the differences when adapting Transformer from language to vision, Liu *et al.* (2021) develop a general-purpose Transformer backbone that computes hierarchical feature map representations with Shifted windows (Swin). Specifically, starting from small-sized patches, Swin Transformer gradually merges neighboring patches going into deep layers. These hierarchical feature maps allow the Swin Transformer to conveniently incorporate multiscale architectures such as FPN (Lin *et al.*, 2017b) or U-Net (Ronneberger *et al.*, 2015). It achieves linear computational complexity through computing self-attention within local non-overlapping windows that contains fixed number of patches and evenly partition an image. The Swin approach alternates partitioning configurations in consecutive layers, establishing cross-window connections that enhance modeling power. The Swin Transformer can be used as backbones for many vision tasks including image classification and object detection. It also achieves state-of-the-art performance on COCO (Lin *et al.*, 2014) and ADE20K (Zhou *et al.*, 2017) datasets in the semantic segmentation task.

CCNet

Huang *et al.* (2019c) propose a different way to reduce the computational complexity of using attention mechanism. Instead of creating dense connections to capture pixel-wise contextual information, they elect to connect each position in the feature map to only the other positions that are in the same row and column, or the “criss-cross” path. They argue that this approach resembles using multiple sparsely-connected

graphs to represent a common single densely-connected graph (Huang *et al.*, 2019c). This strategy effectively reduces both space and time computational complexities from $O(N^2)$ to $O(\sqrt{N})$. This model is able to achieve superior performance on various public semantic segmentation datasets than DeepLab (Yu *et al.*, 2018c; Chen *et al.*, 2018c) models and similar attention-based models including PSANet (Zhao *et al.*, 2018b) and DFN (Yu *et al.*, 2018c).

Hierarchical Multiscale Attention for Semantic Segmentation

Chen *et al.* (2016b) develop an attention mechanism that utilizes multiscale feature maps as input to train attention masks simultaneously at each scale. In contrast, Tao *et al.* (2020) develops a hierarchical attention-based approach that learns relative attention mask between adjacent scales, which allows for faster training with pairs of images. Using the HRNet-OCR (Yuan *et al.*, 2020b) backbone network, it also achieves state-of-the-art performance on Cityscapes (Cordts *et al.*, 2016) dataset.

MaskFormer

Cheng *et al.* (2021) propose the MaskFormer model based on the observation that semantic- and instance-level segmentation can be solved in a unified manner with mask classification using the same model, loss, and training procedure. MaskFormer performs simple mask classification with three modules. The pixel-level module generates per-pixel embeddings using the feature map output from a backbone CNN. The transformer module computes pairs of class predictions and mask embedding vectors, each corresponding to a semantic class. Finally, the segmentation module computes the dot product between the mask embedding vectors and the per-pixel embeddings followed by sigmoid activation to produce binary mask predictions. These binary mask predictions are combined through matrix multiplication to produce final semantic segmentation labels. MaskFormer also naturally solves panoptic segmentation and achieves superior performance for both tasks on ADE20K (Zhou *et al.*, 2017) and COCO (Lin *et al.*, 2014) benchmark datasets.

Mask2Former

Mask2Former (Cheng *et al.*, 2022a) builds on MaskFormer and makes three improvements. First, standard cross-attention in the Transformer decoder is replaced with masked attention which limits the attention to localized features centered around predicted segments instead of the full feature map (Cheng *et al.*, 2022a). This improves both the efficiency and the performance of the model. Second, high-resolution features are utilized efficiently via a feature pyramid that feeds one scale of the multi-scale feature to one Transformer decoder layer at a time (Cheng *et al.*, 2022a). Finally, reduced computations are achieved by calculating mask loss on K randomly sampled points instead of the entire mask (Cheng *et al.*, 2022a).

Additional Works

Attention-based models have become increasingly popular due to their ability to capture contextual information in a more efficient manner. Other attention-based semantic segmentation models include Strudel *et al.* (2021), Fu *et al.* (2020), Yin *et al.* (2020), and Li *et al.* (2018a), and Yuan *et al.*, 2021b.

3.3.4 Segmentation with Graphical Models

CRF can be used to refine coarse feature map output from FCN, as demonstrated in the DeepLab (Chen *et al.*, 2017a) model. It incorporates CRFs as a post-processing module that significantly boosts its prediction accuracy. However, this mechanism has three major drawbacks: 1) CRF uses some hard-coded parameters that can be difficult to adapt to different test sets; 2) these parameters cannot be trained jointly with FCN; and 3) Incorporation of CRFs significantly increases the computational complexity of the models during inference time. A few models discussed in this section resolve some of these issues through integrating CRFs into the CNN layers to achieve end-to-end training. This improves model efficiency as well as achieving better segmentation performance.

CRF-RNN

Zheng *et al.* (2015) propose a method that formulates CRF as Recurrent Neural Network (RNN), which is integrated with FCN so that the network can be trained from end-to-end. Specifically, the energy function described in Krähenbühl and Koltun (2011) is used to optimize the CRF for semantic label prediction. This energy function consists of Unary energy – which is obtained from a CNN and predicts pixel labels independent of assignment consistency and smoothness, and Pairwise energy – which uses a smoothing term to improve label consistency in similar pixels. The mean-field algorithm is introduced in Kendall *et al.* (2017) to approximate the CRF distribution. Zheng *et al.* (2015) first formulates a single iteration of the mean-field algorithm as a stack of CNNs (Figure 3.6), and then further formulates repeated iterations as an RNN whose parameters can be learned through back-propagation through-time (Mozer, 1989). This approach achieved superior performance on the Pascal VOC 2012 (Everingham *et al.*, 2010) challenge compared to state-of-art methods.

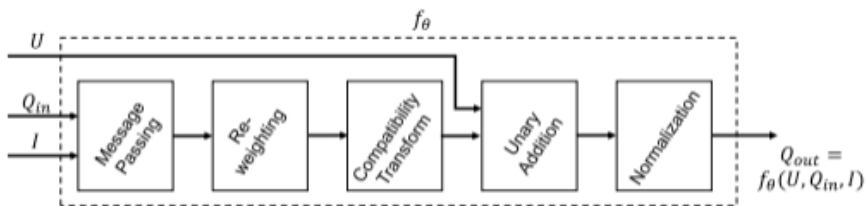


Figure 3.6: A mean-field iteration as a CNN. From Zheng *et al.* (2015).

Piecewise

Unlike Deeplab (Chen *et al.*, 2017a) or CRF-RNN (Zheng *et al.*, 2015), Lin *et al.* (2016) use CRFs to learn ‘patch-patch’ context in image regions. Instead of training CNNs and CRFs jointly, they develop a more efficient piecewise training approach. A node is created in the CRF for each rectangular region in the feature map, and pairwise

connections are established between nodes within a spatial range box (Lin *et al.*, 2016). Different types of range box are created to represent different spatial relations including “surrounding” and “above/below”, and a pairwise potential function is used to model each type of relation. Pairwise potential functions are defined based on the output of Pairwise-Net (Lin *et al.*, 2016), which consists of two fully connected layers that are applied to the edge features that are concatenated feature vectors of two connected nodes. Unary-Net (Lin *et al.*, 2016), which also consists of two fully connected layers, is applied to output from the CNN feature maps to produce the Unary potential functions. Piecewise training of CRFs is done with CNN potentials to allow efficient and parallel training.

DPN

The Deep Parsing Network (DPN) (Liu *et al.*, 2015) is developed to incorporate context information into CRF for semantic segmentation. DPN is based on an extension of the CNN architecture and models unary terms and various types of pairwise terms by approximating the mean field algorithm. DPN achieves high computational efficiency, and its operations can be parallelized. It also achieves competitive performance on the PASCAL VOC 2012 (Everingham *et al.*, 2010) dataset.

GCRF

The Gaussian Mean Field (GMF) network (Vemulapalli *et al.*, 2016) is another approach that incorporates a CRF model and the mean field inference to improve semantic segmentation. This work is different from CRF-RNN (Zheng *et al.*, 2015) in two different ways: first, it uses simpler Gaussian CRFs with reachable optimality, instead of discrete CRFs that are more complex; second, a fixed number of steps of the mean field inference are unrolled as a deep network instead of formulated as RNNs. The proposed GMF network is combined with CNNs to form what the authors term the Gaussian CRF Network (GCRF) (Vemulapalli *et al.*, 2016) that is trainable from end-to-end.

Additional Works

In addition to the work discussed in this section, models from the DeepLab (Chen *et al.*, 2017a) family also use CRF for improving prediction results. Other works include Bell *et al.* (2015) and Fu *et al.* (2016), where a fully connected CRF is used to predict material segmentation in the wild and retina vessel segmentation, respectively.

3.3.5 Models Incorporating Other Forms of Contextual Knowledge

Domain Transform

Despite the enhanced performance CRFs bring to semantic segmentation tasks, dense CRF inference can be demanding on computational power. Chen *et al.* (2016a) propose an approach to replace dense CRFs with a domain transform (DT) edge-preserving filter based on a reference edge map that is directly produced from the encoder CNN that also outputs coarse semantic segmentation scores (Figure 3.7). Specifically, DeepLab (Chen *et al.*, 2017a) is used for semantic segmentation prediction, and EdgeNet (Chen *et al.*, 2016a) is used for edge prediction. DT runs 1-D filters across both rows and columns of the coarse segmentation scores and the reference edge map to calculate the final segmentation scores. This architecture improves boundary pixel predictions while significantly reducing inference time when compared to CRF-based methods (Chen *et al.*, 2017a; Bell *et al.*, 2015; Fu *et al.*, 2016; Vemulapalli *et al.*, 2016; Liu *et al.*, 2015).

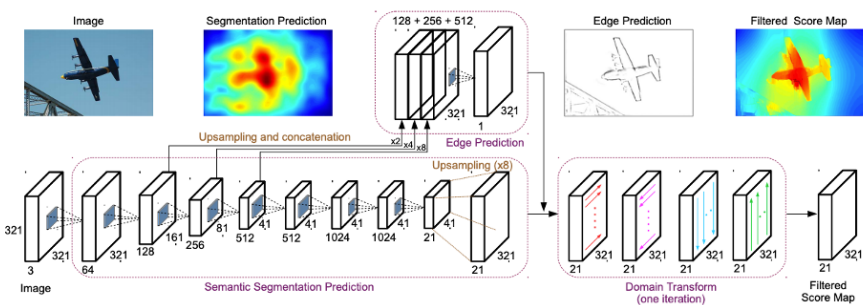


Figure 3.7: Architecture of the proposed Domain Transform approach. From Chen *et al.* (2016a).

GSCNN

Gated Shape CNN (GSCNN) (Takikawa *et al.*, 2019) is a two-stream CNN architecture developed to explicitly incorporate shape information from the image to improve boundary detection for semantic segmentation. A new type of gating mechanism is designed to connect the intermediate layers of the classical and the shape stream. Specifically, higher-level information from the classical stream is passed into the shape stream to denoise activations in early stages. This helps the shape stream to only focus on relevant information. Outputs from the two streams are fused with an ASPP module in multiscale fashion. A dual task regularizer is formulated to capture semantic boundary loss which penalize mismatched boundary pixels. GSCNN outperformed state-of-art approaches such as DeepLab-v3+ (Chen *et al.*, 2018c) on the Cityscapes (Cordts *et al.*, 2016) benchmark dataset.

EncNet

Context Encoding Network (EncNet) (Zhang *et al.*, 2018a) is a framework for semantic segmentation that captures global scene context through a novel Context Encoding Module and Semantic Encoding Loss (SE-loss). The architecture of EncNet is as follows: first, a pretrained ResNet backbone extracts the feature maps from the input image; the Encoding Layer, which is part of the Context Encoding Module, then generates encoded semantics capturing feature statistics and predicts scaling factors for selectively emphasizing class-dependent feature maps; the other part of the Context Encoding Module predicts each object category's presence in the scene based on the encoded semantics and employs the SE-loss that equally considers small and big objects to regularize the training; finally, the representation of the Context Encoding Module is incorporated into the last convolutional layer before EncNet produces the per-pixel predictions. This framework only incurs marginal extra computational costs and achieved new state-of-the-art performance on the PASCAL-Context (Mottaghi *et al.*, 2014) dataset.

Other Models

Contextual Knowledge is a powerful addition to vanilla FCN features in semantic segmentation task, regardless of the form of the knowledge, as seen in the approaches discussed in this section. Other models that leverage contextual information include ACNet (Hu *et al.*, 2019), E-Net (Paszke *et al.*, 2016), Raj *et al.* (2015), Roy and Todorovic (2016), Bian *et al.* (2016), and Pinheiro and Collobert (2014).

3.4 RNN-based Segmentation

In addition to the previously discussed CRF-RNN (Zheng *et al.*, 2015) model, RNNs have been utilized in semantic segmentation tasks to capture the global contextual information through long-range dependencies. Objects in images usually exhibit certain relationships even when they are far away from each other. The use of RNNs addresses the limitation of CNN-based features where only local contexts are incorporated with smaller kernels. Features produced through RNN layers are usually combined with CNN-based features before being passed into the decoder for dense pixel predictions.

ReSeg

Visin *et al.* (2016) design an architecture called ReSeg based on an ReNet (Visin *et al.*, 2015), a previous model for image classification. ReNet is composed of RNN layers that capture contextual information through both horizontal and vertical sweeps of the image and hidden layer outputs. ReNet layers are stacked on the encoder layer outputs from the pretrained CNNs in ReSeg, resulting in an improved feature map that is fed into an upsampling layer before reaching final semantic predictions.

DAG-RNNs

In a separate work, Shuai *et al.* (2016) incorporate directed acyclic graph RNNs (DAG-RNNs) to leverage long-range contextual information in dense pixel predictions. They argue that chain-structured RNNs do

not capture dependencies among image units, and that undirected cyclic graphs (UCGs) can be used to better capture interactions among image units (Shuai *et al.*, 2016). Furthermore, they approximate UCG's topology with multiple directed acyclic graphs (DAGs) since RNNs cannot be applied directly to UCG-structured images. Figure 3.8 shows an example of such approximation. DAG-RNNs are integrated after the encoder CNN layers to model dependencies among the elements in the output feature map. The refined feature map is then passed into the deconvolution layers and up-sampled to dense pixel predictions.

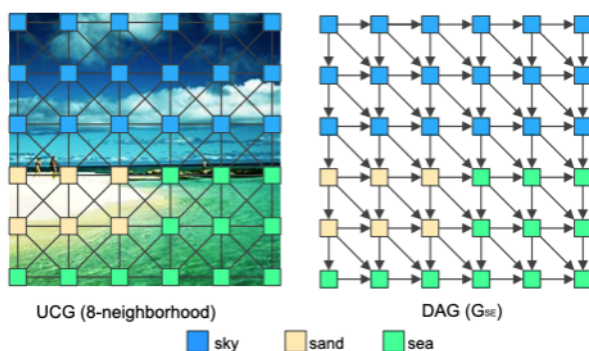


Figure 3.8: An 8-neighborhood UCG and one of its induced DAG in the southeastern (SE) direction. From Shuai *et al.* (2016).

LSTM-based Models

Apart from traditional RNNs, Long Short Term Memory (LSTM) networks have also been applied to semantic segmentation tasks. In Byeon *et al.* (2015), the authors incorporate **2D LSTM** networks to extract in a single process both global and local dependencies from images for scene labeling. The input image is first divided into a $n \times n$ grid. The authors argue that window-based input units maintain global coherence of the image and local correlation of the pixels while reducing computational load. Four LSTM memory blocks which function as a 2D LSTM layer are applied to each window. The windows are connected to each surrounding directions and propagate contexts. A feedforward layer then sums up outputs from each LSTM block and applies a Hyperbolic

tangent ($\tanh$) activation. Outputs from the final LSTM blocks are summed before fed into the softmax layer to obtain probabilistic label output for each input window. This approach achieves competitive performance on Stanford Background (Gould *et al.*, 2009) and SIFT Flow (Liu *et al.*, 2011a) datasets with much computational complexity. However, it suffers from mistakes of mislabeling well-segmented regions due to some properties of the LSTM networks (Byeon *et al.*, 2015).

Li *et al.* (2016) integrate a novel LSTM structure named **LSTM-CF** into a CNN to capture contextual information from photometric and depth data. This allows end-to-end training of the network for semantic RGB-D label predictions. It also incorporates 2D contexts with both vertical and horizontal information.

In Liang *et al.* (2016), **Graph LSTM** Network is used to improve the modeling of the visual patterns such as object boundaries and appearance similarities by more effectively propagating information among pixels based on graph-structured representations. An undirected graph is adaptively constructed based on superpixels obtained using SLIC (Achanta *et al.*, 2010). Edges naturally represent spatial relations of the superpixels (Liang *et al.*, 2016), and Confidence-Driven Search (CDS) is used to update the states of all nodes (Liang *et al.*, 2016). Additionally, each graph LSTM unit's forget gate learns to incorporate local contextual interactions among neighboring nodes. This Graph LSTM structure effectively exploits global context to boost local predictions, achieving superior performance on semantic object parsing challenge datasets including PASCAL-Person-Part dataset (Chen *et al.*, 2014), Horse-Cow parsing dataset (Wang and Yuille, 2015), ATR dataset (Liang *et al.*, 2015) and Fashionista dataset (Yamaguchi *et al.*, 2012). Figure 3.9 shows the architecture of this Graph LSTM Network.

In Hu *et al.* (2016), an approach combining LSTMs and CNNs is used to address the novel task of segmenting images based on linguistic descriptions. The LSTM encodes the natural language expression into a vector representation, and a FCN extracts a spatial feature map from the image (Hu *et al.*, 2016). These two outputs are passed into a fully convolutional classification network and an upsampling network to produce a final pixel-wise segmentation map. These networks are trained from end-to-end to jointly process image and text information.

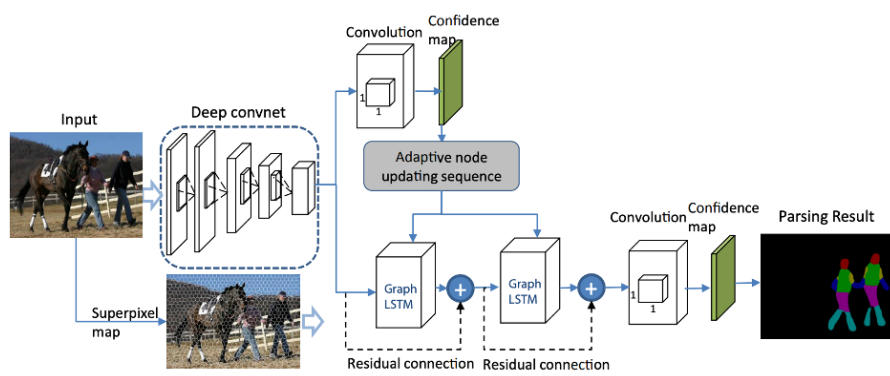


Figure 3.9: The Graph LSTM structure. From Liang *et al.* (2016).

3.5 GAN-based Segmentation

Generative Adversarial Networks (GANs) are often used to distinguish ground truth and synthetic images, and are widely used in image generation tasks. Recently, GANs have also shown promising results when used for semantic segmentation. They have been incorporated into segmentation models to serve two main functions: improving prediction quality with the discriminator and producing additional data with the generator in a semi-supervised training framework.

Luc *et al.* (2016) complement convolutional semantic segmentation network with an adversarial network, whose job is to discriminate between the ground truth segmentation maps and the output label maps from the semantic segmentation network. Their objective function includes a conventional multi-class cross-entropy loss and an adversarial term, allowing the two networks to be jointly trained (Figure 3.10).

Souly *et al.* (2017) leverage GANs in a semi-supervised training framework for semantic segmentation. A generator network is designed to generate synthetic data from both noise and class label information from the input image. Given a limited amount of image data with pixel-level labels, a large amount of image data with only image-level class labels, and the fake image data from the generator network, a discriminator network is trained to produce pixel-level confidence maps for each semantic class and assign fake class labels to the generated

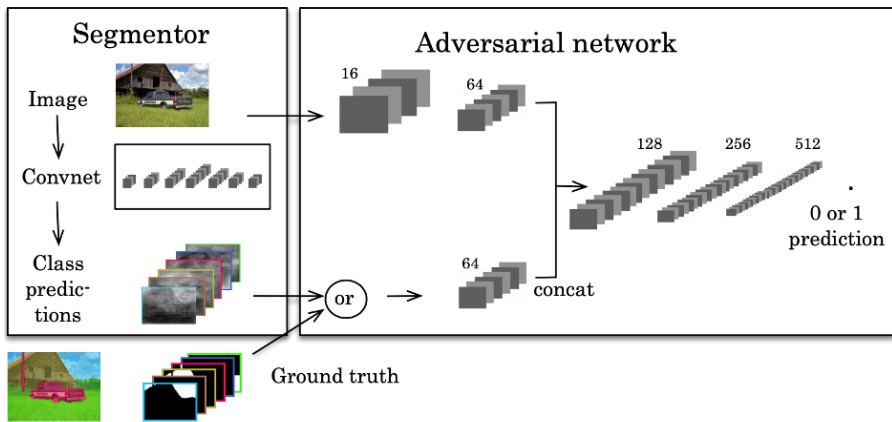


Figure 3.10: Overview of the proposed approach by Luc *et al.* (2016).

samples. A novel loss function different from traditional GANs is designed to handle these three types of input data – it includes a term for pixels from labeled data to be classified as one of the available classes, a term for decreasing the probability of pixels from unlabeled data to be classified as a fake class, and a term for distinguishing real image data from fake samples produced by the generator (Souly *et al.*, 2017).

Hung *et al.* (2018) use another semi-supervised learning approach for improving semantic segmentation. The segmentation network is trained by minimizing a multitask loss composed of a cross-entropy loss, an adversarial loss, and a semi-supervised loss. The standard cross-entropy loss is computed for the ground truth and the predicted segmentation map. The predicted segmentation map is passed through a discriminator to produce a confidence map, which is used to semi-supervise the segmentation network during training by providing signals on the quality of predictions. In addition, the segmentation network is supervised by the adversarial loss from the discriminator network, which allows the segmentation network to learn to produce from unlabeled data segmentation outputs with distributions close to that of the ground truth.

SegAN (Xue *et al.*, 2018) is a framework designed for medical image segmentation. It includes a segmentor and a critic network trained in

an alternating fashion. The segmentor S is a fully convolutional neural network with skip connections. The critic C is trained by maximizing a multiscale L1 loss function between predicted and ground truth segmentation. S aims to minimize the same objective function. During training, S is first fixed while C is trained for one step with gradients from the loss function, and C is fixed while S is trained with the gradients passed from C from the same loss function. These two steps alternate to train the S and C networks in an adversarial fashion. The multiscale feature loss enforces the networks to learn hierarchical features by capturing long- and short-range spatial relations between pixels (Xue *et al.*, 2018). This approach achieves competitive performance on the BRATS brain tumor segmentation (Menze *et al.*, 2014) dataset, outperforming U-net (Ronneberger *et al.*, 2015).

In Chartsias *et al.* (2017), cardiac MR images from CT images were generated with a conditional GAN to boost segmentation accuracy. Similar works leveraging GAN for synthetic image generation include Zhang *et al.* (2018c), Shin *et al.* (2018), Changhee *et al.* (2019), Yang *et al.* (2020c), Yu *et al.* (2018a), and Abhishek and Hamarneh (2019). These methods are widely applied to medical image segmentation, but have also been used for natural images such as remote sensing imagery (Mohajerani *et al.*, 2019).

3.6 Meta-learning for Network Structures

Design of network architectures has significant impact on its performance. Many works discussed in this review are based on human-invented architectures, but recently, meta-learning methods have shown promising results for image classification tasks. These methods produce network structures dynamically by conducting a graph search. Various graph structures and search algorithms have been designed to optimize the efficiency of the process and the structure of the resulting network. Given sufficient computational resources, these methods have the potential to generate novel and powerful network architectures that can surpass human designs and achieve superior predictive accuracy.