Understanding Deep Learning Techniques for Image Segmentation

SWARNENDU GHOSH, NIBARAN DAS, ISHITA DAS, and UJJWAL MAULIK,
Jadavpur University

The machine learning community has been overwhelmed by a plethora of deep learning–based approaches. Many challenging computer vision tasks, such as detection, localization, recognition, and segmentation of objects in an unconstrained environment, are being efficiently addressed by various types of deep neural networks, such as convolutional neural networks, recurrent networks, adversarial networks, and autoencoders. Although there have been plenty of analytical studies regarding the object detection or recognition domain, many new deep learning techniques have surfaced with respect to image segmentation techniques. This article approaches these various deep learning techniques of image segmentation from an analytical perspective. The main goal of this work is to provide an intuitive understanding of the major techniques that have made a significant contribution to the image segmentation domain. Starting from some of the traditional image segmentation approaches, the article progresses by describing the effect that deep learning has had on the image segmentation domain. Thereafter, most of the major segmentation algorithms have been logically categorized with paragraphs dedicated to their unique contribution. With an ample amount of intuitive explanations, the reader is expected to have an improved ability to visualize the internal dynamics of these processes.

CCS Concepts: • **Computing methodologies** → **Image segmentation**; **Neural networks**;

Additional Key Words and Phrases: Deep learning, semantic image segmentation, convolutional neural networks

ACM Reference format:
Swarnendu Ghosh, Nibaran Das, Ishita Das, and Ujjwal Maulik. 2019. Understanding Deep Learning Techniques for Image Segmentation. *ACM Comput. Surv.* 52, 4, Article 73 (August 2019), 35 pages.
https://doi.org/10.1145/3329784

1 INTRODUCTION

Image segmentation can be defined as a specific image processing technique that is used to divide an image into two or more meaningful regions. Image segmentation can also be seen as a process of defining boundaries between separate semantic entities in an image. From a more technical perspective, image segmentation is a process of assigning a label to each pixel in the image such

This work was partially supported by project order no. SB/S3/EECE/054/2016, dated November 25, 2016, sponsored by SERB (Government of India), and carried out at the Centre for Microprocessor Application for Training Education and Research, CSE Department, Jadavpur University.
Authors' addresses: S. Ghosh, N. Das, I. Das, and U. Maulik, Jadavpur University, Kolkata, WB, 700032, India; emails: swarbir@gmail.com, nibarandas@jadavpuruniversity.in, ishitad123@gmail.com, ujjwal_maulik@yahoo.com.

0360-0300/2019/08-ART73 $15.00
https://doi.org/10.1145/3329784

Fig. 1. Semantic image segmentation (samples from the Mapillary Vistas dataset [126]).

that pixels with the same label are connected with respect to some visual or semantic property (Figure 1).

Image segmentation subsumes a large class of finely related problems in computer vision. The most classic version is semantic segmentation [53]. In semantic segmentation, each pixel is classified into one of the predefined set of classes such that pixels belonging to the same class belong to a unique semantic entity in the image. It is also worthy to note that the semantics in question depend not only on the data but also the problem that needs to be addressed. For example, for a pedestrian detection system, the whole body of person should belong to the same segment; however, for an action recognition system, it might be necessary to segment different body parts into different classes. Other forms of image segmentation can focus on the most important object in a scene. A particular class of problem called *saliency detection* [14] is born from this. Other variants of this domain can be foreground-background separation problems. In many systems, such as image retrieval or visual question answering, it is often necessary to count the number of objects. Instance-specific segmentation addresses that issue. Instance-specific segmentation is often coupled with object detection systems to detect and segment multiple instances of the same object [35] in a scene. Segmentation in the temporal space is also a challenging domain and has various applications. In object tracking scenarios, pixel-level classification is not only performed in the spatial domain but also across time. Other applications in traffic analysis or surveillance need to perform motion segmentation to analyze paths of moving objects. In the field of segmentation with a lower semantic level, over-segmentation is also a common approach where images are divided into extremely small regions to ensure boundary adherence, at the cost of creating a lot of spurious edges. Over-segmentation algorithms are often combined with region merging techniques to perform image segmentation. Even simple color or texture segmentation also finds its use in various scenarios. Another important distinction between segmentation algorithms is the need for interactions from the user. Although it is desirable to have fully automated systems, a little bit of interaction from the user can improve the quality of segmentation to a large extent. This is especially applicable when we are dealing with complex scenes or we do not possess an ample amount of data to train the system.

Segmentation algorithms have several applications in the real world. In medical image processing [99], we also need to localize various abnormalities such as aneurysms [39], tumors [118], cancerous elements like melanoma detection [157], or specific organs during surgeries [172]. Another domain where segmentation is important is surveillance. Many problems, such as pedestrian detection [89] and traffic surveillance [49], require the segmentation of specific objects (e.g.,

persons or cars). Other domains include satellite imagery [9, 12], guidance systems in defense [95], and forensics (e.g., face [5], iris [41], and fingerprint [117] recognition). Generally, traditional methods, such as histogram thresholding [162], hybridization [69, 161] feature space clustering [32], region-based approaches [48], edge detection approaches [152], fuzzy approaches [31], entropy-based approaches [38], neural networks (Hopfield neural network [28], self-organizing maps [20]), and physics-based approaches [129], are used popularly in this purpose. However, such feature-based approaches have a common bottleneck that they are dependent on the quality of feature extracted by the domain experts. Generally, humans are bound to miss latent or abstract features for image segmentation. Yet deep learning in general addresses this issue of automated feature learning. In this regard, one of the most common techniques in computer vision was introduced by the name of convolutional neural networks (CNNs) [87] that learned a cascaded set of convolutional kernels through back-propagation [150]. Since then, it has been improved significantly, with features such as layerwise training [11], rectified linear activations [124], batch normalization [66], auxiliary classifiers [42], atrous convolutions [177], skip connections [63], and better optimization techniques [78]. In addition, there have been a large number of new types of image segmentation techniques. Various such techniques have drawn inspiration from popular networks such as AlexNet [82], convolutional autoencoders [115], recurrent neural networks (RNNs) [116], and residual networks [63].

2 MOTIVATION

There have been many reviews and surveys regarding the traditional technologies associated with image segmentation [50, 131]. Although some of them have specialized in application areas [84, 99, 153], others have focused on specific types of algorithms [14, 15, 48]. With the arrival of deep learning techniques, many new classes of image segmentation algorithms have surfaced. Earlier studies [185] have shown the potential of deep learning–based approaches. There have been more recent studies [55] that cover several methods and compare them on the basis of their reported performance. The work of Garcia-Garcia et al. [53] lists a variety of deep learning–based segmentation techniques. They have tabulated the performance of various state-of-the-art networks on several modern challenges. The resources are incredibly useful for understanding the current state of the art in this domain. Knowing the available methods is quite useful to develop products; however, to contribute to this domain as a researcher, one needs to understand the underlying mechanics of the methods that make them confident. In the present work, our main motivation is to answer the question of why the methods are designed in the way they are. Understanding the mechanics of modern techniques would make it easier to tackle new challenges and develop better algorithms. Our approach carefully analyzes each method to understand why the methods succeed at what they do and also why they fail in certain situations. Being aware of the pros and cons of such methods, new designs can be initiated that reap the benefits of the pros and overcome the cons. We recommend the work of Garcia-Garcia [53] for an overview of some of the best image segmentation techniques using deep learning, whereas our focus is to understand why, when, and how these techniques perform in various challenges.

2.1 Contribution

The article has been designed in a way such that new researchers reap the most benefit. Initially, some of the traditional techniques have been discussed to uphold the frameworks before the deep learning era. Gradually, the various factors governing the onset of deep learning has been discussed so that readers have a good idea of the current direction in which machine learning is progressing. In subsequent sections, the major deep learning algorithms have briefly been described in a generic way to establish a clearer concept of the procedures in the mind of the readers. The

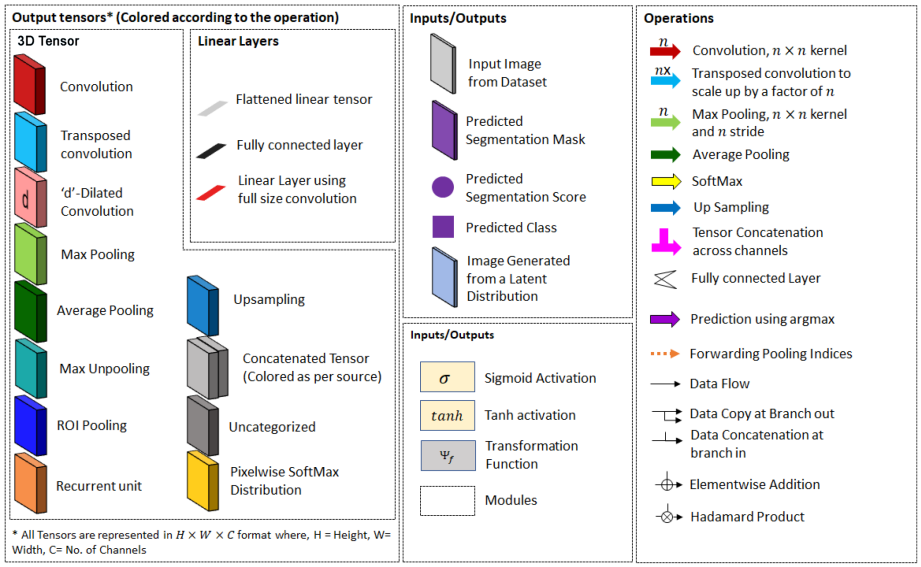

Fig. 2. Legends for subsequent diagrams of popular deep learning architectures.

image segmentation algorithms discussed thereafter have been categorized into the major families of algorithms that have governed the past few years in this domain. The concepts behind all of the major approaches have been explained through a very simple language with a minimum amount of complicated mathematics. Almost all of the diagrams corresponding to major networks have been drawn using a common representational format as shown in Figure 2. The various approaches that have been discussed come with different representations for architectures. The unified representation scheme allows the user to understand the fundamental similarities and differences between networks. Finally, the major application areas have been discussed to help new researchers pursue a field of their choice.

3 IMPACT OF DEEP LEARNING ON IMAGE SEGMENTATION

The development of deep learning algorithms like CNNs or deep autoencoders not only affect typical tasks like object classification but are also efficient in other related tasks like object detection, localization, and tracking, or, as in this case, image segmentation.

3.1 Effectiveness of Convolutions for Segmentation

As an operation, convolution can be simply defined as the function that performs a sum of product between kernel weights and input values while convoluting the smaller kernel over a larger image. For a typical image with k channels, we can convolute a smaller-size kernel with k channels along the x and y direction to obtain an output in the format of a 2D matrix. It has been observed that after training a typical CNN, the convolutional kernels tend to generate activation maps with respect to certain features of the objects [180]. Given the nature of activations, it can be seen as segmentation masks of object-specific features. Hence, the key to generating requirement-specific segmentation is already embedded within these output activation matrices. Most of the image segmentation algorithms use this property of CNNs to somehow generate the segmentation masks as required to solve the problem. As shown in Figure 3, the earlier layers capture local features like the contour or a small part of an object. In the later layers, more global features are activated, such as field,

Fig. 3. Input image and sample activation maps from a typical CNN. (Top row) Input image and two activation maps from earlier layers showing part objects like t-shirts and features like contours. (Bottom row) Activation maps from later layers with more meaningful activations like fields, people, and sky, respectively.

people, or sky. It can also be noted from the figure that the earlier layers show sharper activations compared to the later ones.

3.2 Impact of Larger and More Complex Datasets

The second impact that deep learning has brought to the world of image segmentation is the plethora of datasets, challenges, and competitions. These factors have encouraged researchers across the world to come up with various state-of-the-art technologies to implement segmentation across various domains. A list of many such datasets has been provided in Table 1.

4 IMAGE SEGMENTATION USING DEEP LEARNING

As explained earlier, convolutions are quite effective in generating semantic activation maps that have components that inherently constitute various semantic segments. Various methods have been implemented to make use of these internal activations to segment the images. A summary of major deep learning–based segmentation algorithms are provided in Table 2, along with a brief description of their major contribution.

4.1 Convolutional Neural Networks

CNNs, being one of the most commonly used methods in computer vision, have adopted many simple modifications to perform well in segmentation tasks.

4.1.1 Fully Convolutional Layers. Classification tasks generally require a linear output in the form of a probability distribution over the number of classes. To convert volumes of 2D activation maps into linear layers, they often were flattened. The flattened shape allowed the execution of fully connected networks to obtain the probability distribution. However, this kind of reshaping loses the spatial relations among the pixels in the image. In a fully convolutional neural network (FCN) [105], the output of the last convolutional block is directly used for a pixel-level

Table 1. List of Various Datasets in the Image Segmentation Domain

Category	Dataset
Natural scenes	Berkeley Segmentation Dataset [114]
	PASCAL VOC [44]
	Stanford Background Dataset [58]
	Microsoft COCO [98]
	MIT Scene Parsing Data (ADE20K) [81, 119]
	Semantic Boundaries Dataset [60]
	Microsoft Research Cambridge Object Recognition Image Database (MSRC) [156]
Video segmentation dataset	Densely Annotated Video Segmentation (DAVIS) [138]
	Video Segmentation Benchmark (VSB100) [51]
	YouTube-Video Object Segmentation [175]
Autonomous driving	Cambridge-Driving Labeled Video Database (CamVid) [16]
	Cityscapes: Semantic Urban Scene Understanding [33]
	Mapillary Vistas Dataset [126]
	SYNTHIA: Synthetic Collection of Imagery and Annotations [148]
	KITTI Vision Benchmark Suite [54]
	Berkeley Deep Drive [178]
	India Driving Dataset (IDD) [168]
Aerial imaging	Inria Aerial Image Labeling Dataset [109]
	Aerial Image Segmentation Dataset [179]
	ISPRS Dataset Collection [46]
	Google Open Street Map [7]
	DeepGlobe [40]
Medical imaging	DRIVE: Digital Retinal Images for Vessel Extraction [159]
	Sunnybrook Cardiac Data [141]
	Multiple Sclerosis Database [18, 104]
	IMT: Intima Media Thickness Segmentation Dataset [121]
	SCR: Segmentation in Chest Radiographs [167]
	BRATS: Brain Tumor Segmentation [119]
	LITS: Liver Tumour Segmentation [59]
	BACH: Breast Cancer Histology [6]
	IDRiD: Indian Diabetic Retinopathy Image Dataset [139]
	ISLES: Ischemic Stroke Lesion Segmentation [110]
Saliency detection	MSRA Salient Object Database [30]
	ECSSD: Extended Complex Scene Saliency Dataset [155]
	PASCAL-S Dataset [93]
	THUR15K: Group Saliency in Image [29]
	JuddDB: MIT Saliency Benchmark [13]
	DUT-OMRON Image Dataset [176]
Scene text segmentation	KAIST Scene Text Database [88]
	COCO-Text [169]
	SVT: Street View Text Dataset [171]

Table 2. Summary of Major Deep Learning–Based Segmentation Algorithms

Method	Year	Supervision					Learning			Type			Modules		Description
		S	W	U	I	P	SO	MO	AD	SM	CL	IN	RNN	E-D	
Global Average Pooling	2013		✓				✓				✓				Object-specific soft segmentation
DenseCRF	2014					✓	✓			✓					Using CRF to boost segmentation
FCN	2015	✓					✓			✓					Fully convolutional layers
DeepMask	2015	✓						✓			✓				Simultaneous learning for segmentation and classification
U-Net	2015	✓					✓			✓				✓	Encoder-decoder with multi-scale feature concatenation
SegNet	2015	✓					✓			✓				✓	Encoder-decoder with forwarding pooling indices
CRF as RNN	2015	✓						✓		✓			✓		Simulating CRFs as trainable RNN modules
Deep Parsing Network	2015	✓						✓							Using unshared kernels to incorporate higher-order dependency
BoxSup	2015		✓							✓					Using bounding box for weak supervision
SharpMask	2016	✓						✓			✓			✓	Refined DeepMask with multi-layer feature fusion
Attention to Scale	2016	✓					✓			✓					Fusing features from multi-scale inputs
Semantic Segmentation	2016	✓							✓	✓					Adversarial training for image segmentation
Conv LSTM and Spatial Inhibition	2016	✓						✓				✓	✓		Using spatial inhibition for instance segmentation
JULE	2016			✓				✓		✓			✓		Joint unsupervised learning for segmentation
ENet	2016	✓					✓			✓					Compact network for real-time segmentation
Instance-Aware Segmentation	2016	✓						✓				✓			Multi-task approach for instance segmentation
Mask R-CNN	2017	✓						✓		✓					Using region proposal network for segmentation
Large Kernel Matters	2017	✓					✓			✓				✓	Using larger kernels for learning complex features
RefineNet	2017	✓					✓			✓				✓	Multi-path refinement module for fine segmentation
PSPNet	2017	✓					✓			✓					Multi-scale pooling for scale-agnostic segmentation
Tiramisu	2017	✓					✓			✓				✓	DenseNet 121-feature extractor
Image-to-Image Translation	2017	✓							✓	✓				✓	Conditional GAN for translation image to segment maps
Instance Segmentation with Attention	2017	✓						✓				✓	✓		Attention modules for image segmentation
W-Net	2017			✓				✓		✓				✓	Unsupervised segmentation using normalized cut loss
PolygonRNN	2017				✓		✓			✓			✓		Generating contours by RNN
Deep Layer Cascade	2017	✓					✓			✓					Multi-level approach to handle pixels of different complexity
Spatial Propagation Network	2017	✓					✓			✓					Refinement using linear label propagation
DeepLab	2018	✓					✓			✓					Atrous convolution, spatial pooling pyramid, DenseCRF
SegCaps	2018	✓						✓							Capsule networks for segmentation
Adversarial Collaboration	2018			✓				✓							Adversarial collaboration between multiple networks
Superpixel Supervision	2018			✓				✓							Using super-pixel refinement as supervisory signals
Deep Extreme Cut	2018				✓		✓			✓					Using extreme points for interactive segmentation
Two Stream Fusion	2019				✓		✓			✓					Using image stream and interaction stream simultaneously
SegFast	2019	✓					✓			✓				✓	Using depthwise separable convolution in SqueezeNet encoder

S, supervised; W, weakly supervised; U, unsupervised; I, interactive; P, partially supervised; SO, single-objective optimization; MO, Multi-objective optimization; AD, adversarial learning; SM, semantic segmentation; CL, class-specific segmentation; IN, instance segmentation; RNN, recurrent neural network modules; E-D, encoder-decoder architecture; GAN, generative adversarial network; CRF, conditional random field.

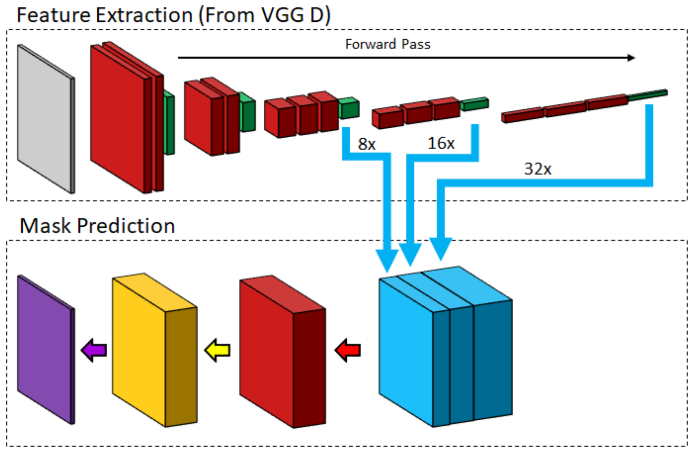

Fig. 4. A fully convolutional network with image segmentation with concatenated multi-scale features.

classification. FCNs were first implemented on the PASCAL VOC 2011 segmentation dataset [44] and achieved a pixel accuracy of 90.3% and a mean IOU of 62.7%. Another way to avoid fully connected linear layers is the use of a full-size average pooling to convert a set of 2D activation maps to a set of scalar values. As these pooled scalars are connected to the output layers, the weights corresponding to each class may be used to perform weighted summation of the corresponding activation maps in the previous layers. This process, called *Global Average Pooling* (GAP) [97], can be directly used on various trained networks, such as residual networks, to find object-specific activation zones that can be used for pixel-level segmentation. The major issues with an algorithm such as this is the loss of sharpness due to the intermediate sub-sampling operations. Sub-sampling is a common operation in CNNs to increase the sensory area of kernels. What it means is that as the activations maps reduce in size in the subsequent layers, the kernels convoluting over them actually correspond to a larger area in the original image. However, it reduces the image size in the process, which when up-sampled to the original size loses sharpness. Many approaches have been implemented to handle this issue. For fully convolutional models, skip connections from preceding layers can be used to obtain sharper versions of the activations from which finer segments can be chalked out (Figure 4). Another work showed how the use of high-dimensional kernels to capture global information with FCN models created better segmentation masks [135]. Segmentation algorithms can also be treated as a boundary detection technique. Convolutional features are also very useful from that perspective [113]. Whereas earlier layers can provide fine details, later layers focus more on the coarser boundaries.

DeepMask and SharpMask. DeepMask [136] was a name given to a project at Facebook AI Research (FAIR) related to image segmentation. It exhibited the same school of thought as FCN models, except the model was capable of multi-tasking (Figure 5). It had two main branches coming out of a shared feature representation. One of them created a pixel-level classification of or a probabilistic mask for the central object, and the second branch generated a score corresponding to the object recognition accuracy. The network was coupled with sliding windows of 16 strides to create segments of objects at various locations of the image, whereas the score helped in identifying which of the segments were good. The network was further upgraded in SharpMask [137], where probabilistic masks from each layer were combined in a top-down fashion using convolutional refinements at every step to generate high-resolution masks (Figure 6). SharpMask scored an

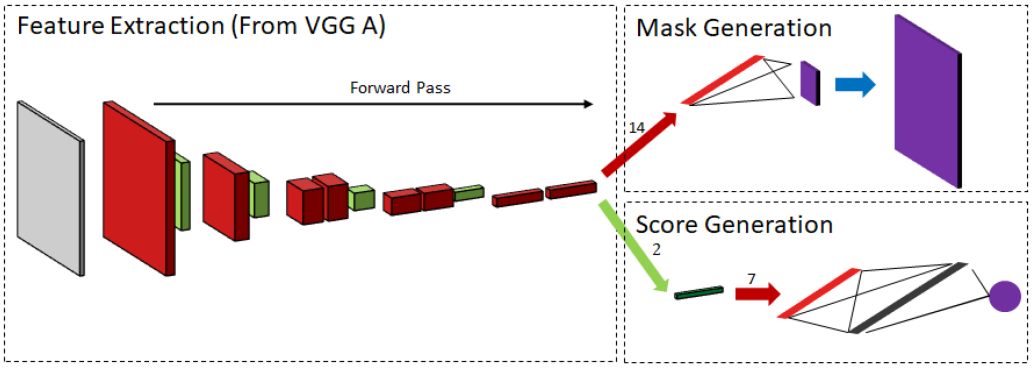

Fig. 5. The DeepMask network.

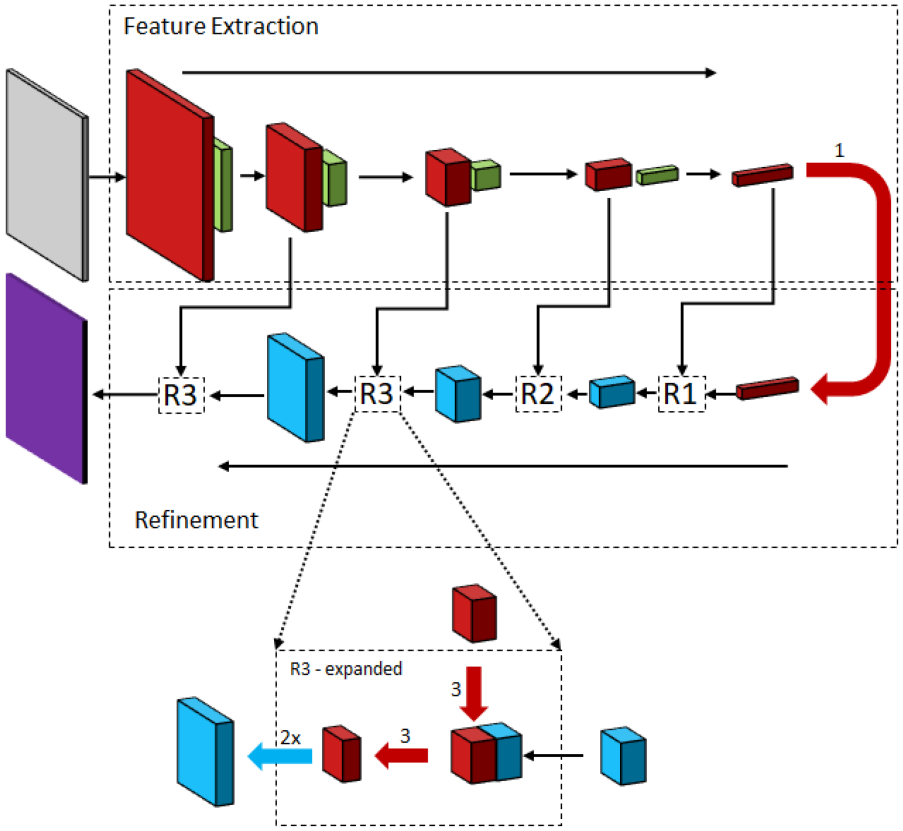

Fig. 6. The SharpMask network.

average recall of 39.3, which beats DeepMask, which scored 36.6 on the MS COCO Segmentation Dataset.

4.1.2 Region Proposal Networks. Another similar wing that started developing with image segmentation was object localization. Task such as this involved locating specific objects in images. Expected outputs for such problems are normally a set of bounding boxes corresponding to the

queried objects. Although strictly stating, some of these algorithms do not address image segmentation problems; however, their approaches are of relevance to this domain.

Region-Based Convolutional Neural Networks. The introduction of the CNNs raised many new questions in the domain of computer vision, one of them primarily being whether a network like AlexNet can be extended to detect the presence of more than one object. Region-based convolutional neural networks [57], more commonly known as R-CNNs, used a selective search technique to propose probable object regions and performed classification on the cropped window to verify sensible localization based on the output probability distribution. The selective search technique [165, 166] analyzes various aspects like texture, color, or intensities to cluster the pixels into objects. The bounding boxes corresponding to these segments are passed through classifying networks to short list some of the most sensible boxes. Finally, with a simple linear regression network, a tighter coordinate can be obtained. The main downside of the technique is its computational cost. The network needs to compute a forward pass for every bounding box proposition. The problem with sharing computation across all boxes was that the boxes were of different sizes, and hence uniform-size features were not achievable. In the upgraded Fast R-CNN [56], Region-of-Interest (ROI) pooling was proposed in which ROIs were dynamically pooled to obtain a fixed-size feature output. Henceforth, the network was mainly bottlenecked by the selective search technique for candidate region proposal. In Faster R-CNN [145], instead of depending on external features, the intermediate activation maps were used to propose bounding boxes, thus speeding up the feature extraction process. Bounding boxes are representative of the location of the object; however, they do not provide pixel-level segments. The Faster R-CNN network was extended as Mask R-CNN [61] with a parallel branch that performed pixel-level object-specific binary classification to provide accurate segments. With Faster R-CNN, an average precision of 35.7 was attained in the COCO [98] test images. The family of RCNN algorithms have been depicted in Figure 7. Region proposal networks have often been combined with other networks [36, 94] to give instance-level segmentations. R-CNN was further improved under the name of HyperNet [79] by using features from multiple layers of the feature extractor. Region proposal networks have also been implemented for instance-specific segmentation. As mentioned earlier, object detection capabilities of approaches like R-CNN are often coupled with segmentation models to generate different masks for different instances of the same object [35].

4.1.3 DeepLab. Although pixel-level segmentation was effective, two complementing issues were still affecting performance. First, smaller kernel sizes failed to capture contextual information. In classification problems, this is handled using pooling layers that increase the sensory area of the kernels with respect to the original image. But in segmentation, that reduces the sharpness of the segmented output. Alternative usage of larger kernels tends to be slower due to significantly larger number of trainable parameters. To handle this issue, the DeepLab [23, 25] family of algorithms (refer Figure 9) demonstrated the use of various methodologies such as atrous convolutions [177], spatial pooling pyramids [62], and fully connected conditional random fields (CRFs) [80] to perform image segmentation with great efficiency. The DeepLab algorithm was able to attain a mean IOU of 79.7 on the PASCAL VOC 2012 dataset [44].

Atrous/Dilated Convolution. The size of the convolution kernels in any layer determine the sensory response area of the network. Whereas smaller kernels extract local information, larger kernels try to focus on more contextual information. However, larger kernels normally come with more parameters. For example, to have a sensory region of 6×6, one must have 36 neurons. To reduce the number of parameters in the CNN, the sensory area is increased in higher layers through techniques like pooling. Pooling layers reduce the size of the image. When an image is pooled by

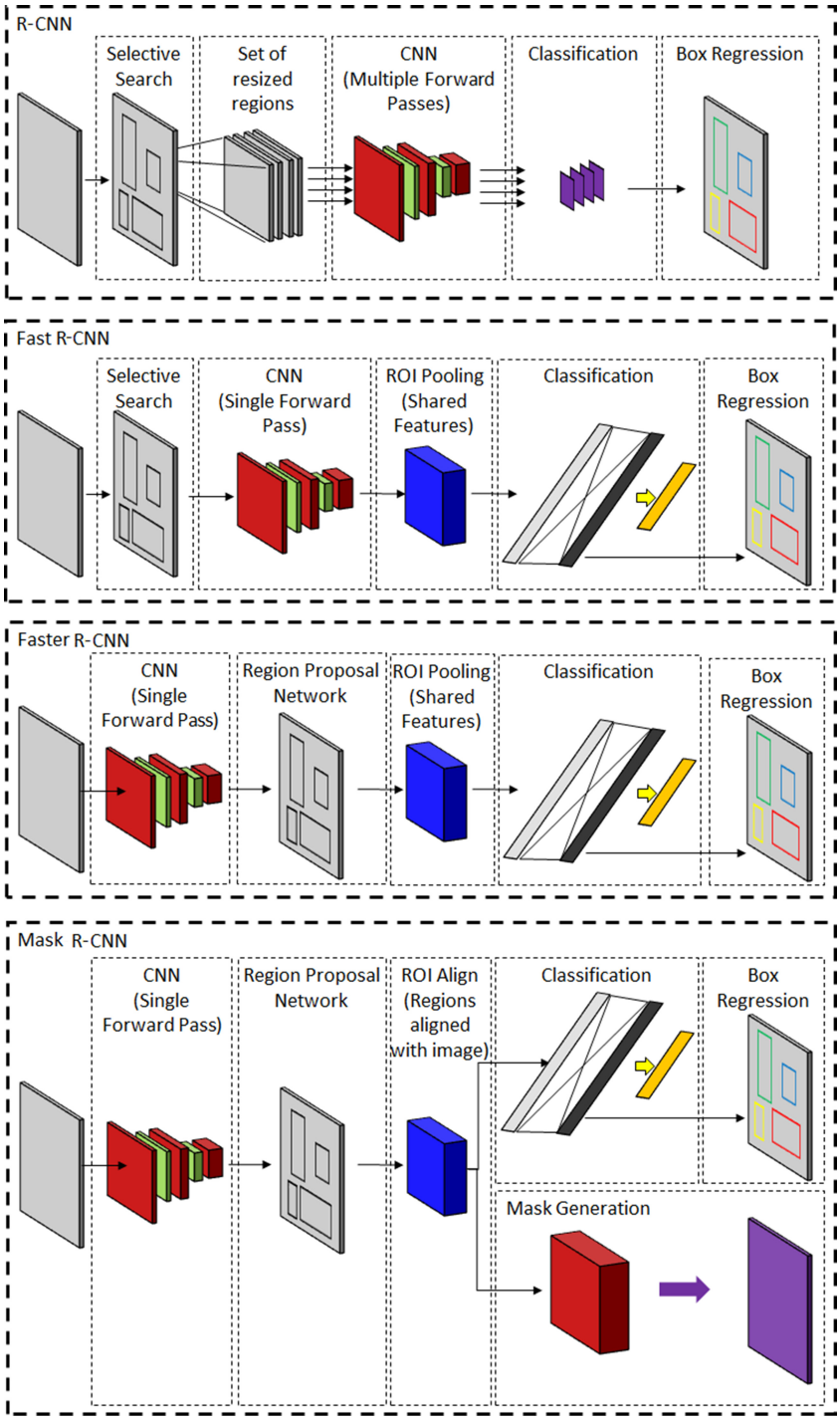

Fig. 7. The R-CNN family of localization and segmentation networks.

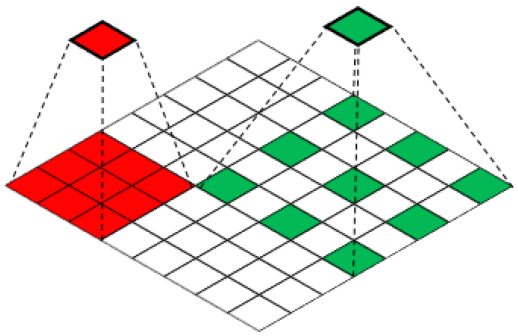

Fig. 8. Normal convolution (red) versus atrous or dilated convolution (green).

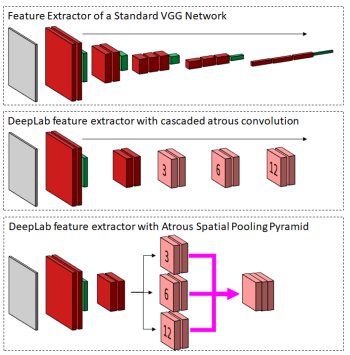

Fig. 9. DeepLab architecture compared to a standard VGG network (top) along with a cascaded atrous convolutions (middle) and atrous spatial pooling pyramid (bottom).

a 2×2 kernel with a stride of 2, the size of the image reduces by 25%. A kernel with an area of 3×3 corresponds to a larger sensory area of 6×6 in the original image. However, unlike before, now only 18 neurons (9 for each layer) are needed in the convolution kernel. In case of segmentation, pooling creates new problems. The reduction in the image size results in loss of sharpness in generated segments as the reduced maps are scaled up to image size. To deal with these two issues simultaneously, dilated or atrous convolutions play a key role. Atrous/dilated convolutions increase the field of view without increasing the number of parameters. As shown in Figure 8, a 3×3 kernel with a dilation factor of 1 can act on an area of 5×5 in the image. Each row and column of the kernel has 3 neurons that are multiplied with intensity values in the image that are separated by the dilation factor of 1. In this way, the kernels can span over larger areas while keeping the number of neurons low and also preserving the sharpness of the image. Besides the DeepLab algorithms, atrous convolutions [27] have also been used with autoencoder-based architectures.

Spatial Pyramid Pooling. Spatial pyramid pooling [62] was introduced in R-CNN, where ROI pooling showed the benefit of using multi-scale regions for object localization. However, in DeepLab, atrous convolutions were preferred over pooling layers for changing the field of view or sensory area. To imitate the effect of ROI pooling, multiple branches with atrous convolutions of different dilations were combined together to utilize multi-scale properties for image segmentation.

Fully Connected CRF. CRF is a undirected discriminative probabilistic graphical model that is often used for various sequence learning problems. Unlike discrete classifiers, while classifying a

sample, it takes into account the labels of other neighboring samples. Image segmentation can be treated as a sequence of pixel classifications. The label of a pixel is not only dependent on its own intensity values but also the values of neighboring pixels. The use of such probabilistic graphical models is often used in the field of image segmentation, and hence it deserves a dedicated section (Section 4.1.4).

4.1.4 Using Inter-Pixel Correlation to Improve CNN-Based Segmentation. The use of probabilistic graphical models such as Markov random fields (MRFs) or CRFs for image segmentation thrived on its own even without the inclusion of CNN-based feature extractors. The CRF or MRF is mainly characterized by an energy function with a unary and a pairwise component.

$$E(x) = \underbrace{\sum_{i} \theta_i(x_i)}_{\text{unary potential}} + \underbrace{\sum_{ij} \theta_{ij}(x_i, x_j)}_{\text{pairwise potential}} \tag{1}$$

Wherease non–deep learning approaches focused on building efficient pairwise potentials like exploiting long-range dependencies, designing higher-order potentials, and exploring contexts of semantic labels, deep learning–based approaches focused on generating a strong unary potentials and using simple pairwise components to boost performance. CRFs have usually been coupled with deep learning–based methods in two ways. One as a separate post-processing module and the other as an trainable module in an end-to-end network like deep parsing networks [103] or spatial propagation networks [101].

Using CRFs to Improve Fully Convolutional Networks. One of the earliest implementations that kickstarted this paradigm of boundary refinements was the work of Krähenbühl and Koltun [81]. With the introduction of fully convolutional networks for image segmentation, it was quite possible to draw coarse segments for objects in images. However, getting sharper segments was still a problem. In the work of Chen et al. [22], the output pixel-level prediction was used as a unary potential for a fully connected CRF. For each pair of pixels i and j in the image, the pairwise potential was defined as in (2).

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \left[w_1 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} \quad -\frac{||I_i - I_j||^2}{2\sigma_\beta^2} \right) + w_2 \exp\left(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2} \right) \right] \tag{2}$$

Here, $\mu(x_i, x_j) = 1$ if $x_i \neq x_j, 0\_\text{otherwise}$ and w_1, w_2 are the weights given to the kernels. The expression uses two Gaussian kernels. The first one is a bilateral kernel that depends on both pixel positions (p_i, p_j) and their corresponding intensities in the RGB channels. The second kernel is only dependent on the the pixel positions. $\sigma_\alpha, \sigma_\beta,$ and σ_γ controls the scale of the Gaussian kernels. The intuition behind the design of such a pairwise potential energy function is to ensure that nearby pixels of similar intensities in the RGB channels are classified under the same class. This model has also been later included in the popular network called *DeepLab* (refer to Section 4.1.3). In the various versions of the DeepLab algorithm, the use of CRF was able to boost the mean IOU on the Pascal 2012 Dataset by a significant amount (up to 4% in some cases).

CRF as RNN. Although CRF is an useful post-processing module [81] for any deep learning–based semantic image segmentation architecture, one of the main drawbacks was that it could not be used as a part of an end-to-end architecture. In the standard CRF model, the pairwise potentials can be represented in terms of a sum of weighted Gaussians. However, since the exact minimization is intractable, a mean-field approximation of the CRF distribution is considered to represent the distribution with a simpler version that simply is a product of independent marginal distributions. This mean-field approximation in its native form is not suitable for back-propagation. In the work

of Zheng et al. [24], this step was replaced by a set of convolutional operations iterated over a recurrent pipeline until convergence was reached. As reported in their work, with the proposed approach, a mean IOU of 74.7 was obtained compared to 71.0 by BoxSup and 72.7 by DeepLab. The sequence of operations can most easily be explained as follows:

(1) *Initialization*: A SoftMax operations over the unary potentials can give us the initial distribution to work with.
(2) *Message Passing*: Convoluting using two Gaussian kernels, one spatial and one bilateral kernel. Similar to the actual implementation of CRF, the splatting and slicing also occurs while building the permutohedral lattice for efficient computation of the fully connected CRF.
(3) *Weighting Filter Outputs*: Convoluting with 1×1 kernels with the required number of channels, the filter outputs can be weighted and summed. The weights can be easily learned through back-propagation.
(4) *Compatibility Transform*: Considering a compatibility function to keep a track of uncertainty between various labels, a simple 1×1 convolution with the same number of input and output channels is enough to simulate that. Unlike the Potts model that assigns the same penalty, here the compatibility function can be learned and hence is a much better alternative.
(5) *Adding the Unary Potentials*: This can be performed by a simple elementwise subtraction of the penalty from the compatibility transform from the unary potentials.
(6) *Normalization*: The outputs can be normalized with another simple softmax function.

Incorporating Higher-Order Dependencies. Another end-to-end network inspired from CRFs incorporates higher-order relations into a deep network. With a deep parsing network [103], pixelwise prediction from a standard VGG-like feature extractor (but with lesser pooling operations) is boosted using a sequence of special convolution and pooling operations. First, by using local convolutions that implement large unshared convolutional kernels across the different positions of the feature map, to obtain translation-dependent features that model long-distance dependencies. Secondly, similar to standard CRFs, a spatial convolution penalizes probabilistic maps based on local label contexts. Finally, with block min pooling that does a pixelwise min pooling across the depth to accept the prediction with the lowest penalty. Similarly, in the work of Liu et al. [101], a row-/columnwise propagation model was proposed the calculated the global pairwise relationship across an image. With a dense affinity matrix drawn from a sparse transformation matrix, coarsely predicted labels were reclassified based on the affinity of pixels.

4.1.5 Multi-Scale Networks. One of the main problems with image segmentation for natural scene images is that the size of the object of interest is very unpredictable. For example, real-world objects may be of different sizes and objects may look bigger or smaller depending on the position of the object and the camera. The nature of a CNN dictates that delicate small-scale features are captured in early layers, whereas as one moves across the depth of the network, the features become more specific for larger objects. For example, a tiny car in a scene has much lesser chance of being captured in the higher layers due to operations like pooling or down-sampling. It is often beneficial to extract information from feature maps of various scales to create segmentations that are agnostic of the size of the object in the image. Multi-scale autoencoder models [26] consider activations of different resolutions to provide image segmentation output.

PSPNet. The pyramid scene parsing network [186] was built on the FCN-based pixel-level classification network. The feature maps from a ResNet-101 network are converted to activations of

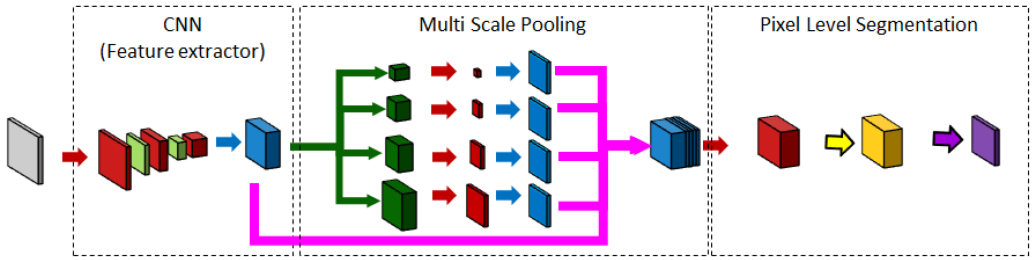

Fig. 10. A schematic representation of PSPNet.

different resolutions thorough multi-scale pooling layers that are later up-sampled and concatenated with the original feature map to perform segmentation (Figure 10). The learning process in deep networks like ResNet was further optimized by using auxiliary classifiers. The different types of pooling modules focus on different areas of the activation map. Pooling kernels of various sizes like 1×1, 2×2, 3×3, and 6×6 look into different areas of the activation map to create the spatial pooling pyramid. On the ImageNet Scene Parsing Challenge, the PSPNet was able to score a mean IoU of 57.21 with respect to 44.80 of FCN and 40.79 of SegNet.

RefineNet. Working with features from the last layer of a CNN produces soft boundaries for the object segments. This issue was avoided in DeepLab algorithms with atrous convolutions. RefineNet [96] takes an alternative approach by refining intermediate activation maps and hierarchically concatenating it to combine multi-scale activations and prevent loss of sharpness simultaneously. The network consisted of separate RefineNet modules for each block of ResNet. Each RefineNet module was made up of three main blocks, namely the residual convolution unit (RCU), multi-resolution fusion (MRF), and chained residual pooling (CRP) (Figure 11). The RCU block consists of an adaptive convolution set that fine tunes the pre-trained weights of the ResNet weights for the segmentation problem. The MRF layer fuses activations of different resolutions using convolutions and up-sampling layers to create a higher-resolution map. Finally, in CRP, layer pooling kernels of multiple sizes are used on the activations to capture the background context from large image areas. RefineNet was tested on the Person-Part Dataset, where it obtained an IOU of 68.6 compared to 64.9 by DeepLab-v2, both of which used ResNet-101 as a feature extractor.

4.2 Convolutional Autoencoders

This section deals with discriminative models that are used to perform pixel-level classification to deal with image segmentation problems. Another line of thought gets its inspiration from autoencoders. Autoencoders have traditionally been used for feature extraction from input samples while trying to retain most of the original information. An autoencoder is basically composed of an encoder that encodes the input representations from a raw input to a possibly lower-dimensional intermediate representation and a decoder that attempts to reconstruct the original input from the intermediate representation. The loss is computed in terms of the difference between the raw input images and the reconstructed output image. The generative nature of the decoder part has often been modified and used for image segmentation purposes. Unlike with traditional autoencoders, during segmentation the loss is computed in terms of the difference between the reconstructed pixel-level class distribution and the desired pixel-level class distribution. This kind of segmentation approach is more of a generative procedure compared to the classification approach of R-CNN or DeepLab algorithms. The problem with approaches such as this is to prevent over-abstraction of images during the encoding process. The primary benefit of such approaches is the ability to generate sharper boundaries with much less complication. Unlike with classification approaches,

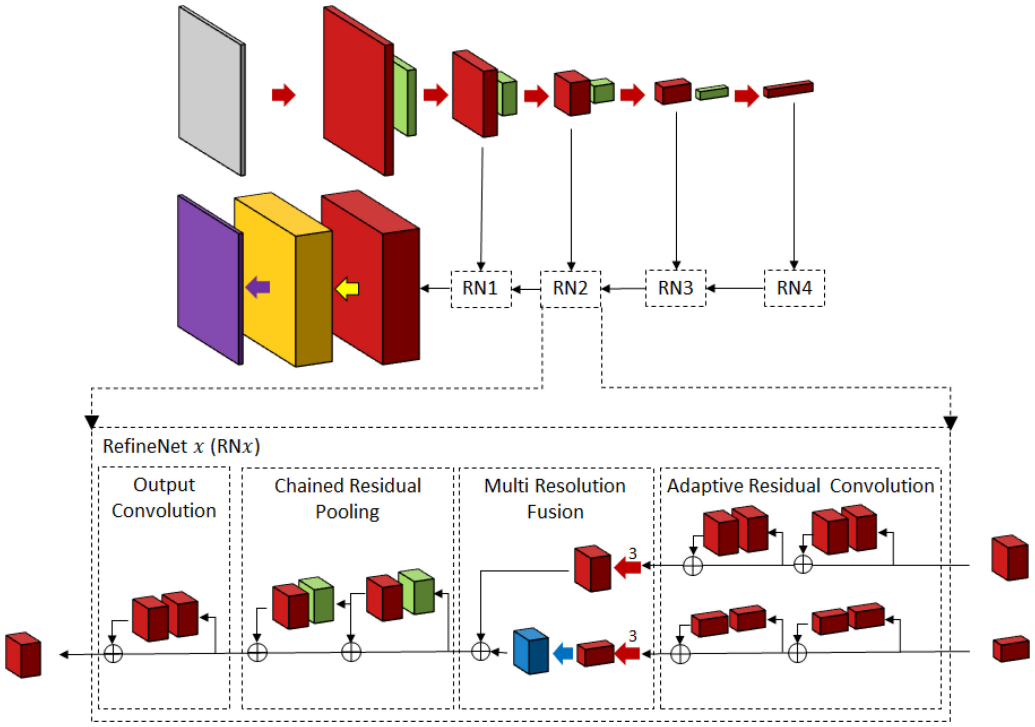

Fig. 11. A schematic representation of the RefineNet.

the generative nature of the decoder can learn to create delicate boundaries based on extracted features. The major issue that affects these algorithms is the level of abstraction. It has been seen that without proper modification, the reduction in the size of the feature map created inconsistencies during the reconstruction. In the paradigm of CNNs, the encoding basically is a series of convolution and pooling layers or strided convolutions. The reconstruction, however, can be tricky. The commonly used techniques for decoding from a lower-dimensional feature are transposed convolution or unpooling layers. One of the main advantages of using an autoencoder-based approach over a normal convolutional feature extractor is the freedom of choosing the input size. With clever use of down-sampling and up-sampling operations, it is possible to output a pixel-level probability that is of the same resolution as the input image. This benefit has made encoder-decoder architectures with multi-scale feature forwarding ubiquitous for networks where the input size is not predetermined and an output of the same size as the input is needed.

Transposed Convolution. Transposed convolution, also known as convolution with fractional strides, has been introduced to reverse the effects of a traditional convolution operation [43, 127]. It is often referred to as deconvolution. However, deconvolution, as defined in signal processing, is different from transposed convolution in terms of the basic formulation, although they effectively address the same problem. In a convolution operation, there is a change in size of the input based on the amount of padding and stride of the kernels. As shown in Figure 12, a stride of 2 will create half the number of activations as that of a stride of 1. For a transposed convolution to work, padding and stride should be controlled in a way that the size change is reversed. This is achieved by dilating the input space. Note that unlike atrous convolutions, where the kernels were dilated, here the input spaces are dilated.

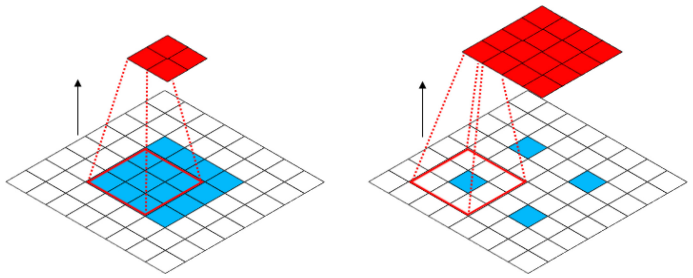

Fig. 12. (Left) Normal convolution with unit stride. (Right) Transposed convolution with fractional strides.

Unpooling. Another approach to reduce the size of the activations is through pooling layers. A 2×2 pooling layer with a stride of 2 reduces the height and width of the image by a factor of 2. In such a pooling layer, a 2×2 neighborhood of pixels is compressed to a single pixel. Different types of pooling perform the compression in different ways. Max pooling considers the maximum activation value among 4 pixels, whereas average pooling takes an average of the same. A corresponding unpooling layer decompresses a single pixel to a neighborhood of 2×2 pixels to double the height and width of the image.

4.2.1 Skip Connections. Linear skip connections have often been used in CNNs to improve gradient flow across a large number of layers [63]. As depth increases in a network, the activation maps tend to focus on more and more abstract concepts. Skip connections have proved to be very effective to combine different levels of abstractions from different layers to generate crisp segmentation maps.

U-Net. The U-Net architecture, proposed in 2015, proved to be quite efficient for a variety of problems, such as segmentation of neuronal structures, radiography, and cell tracking challenges [147]. The network is characterized by an encoder with a series of convolution and max pooling layers. The decoding layer contains a mirrored sequence of convolutions and transposed convolutions. As described until now, it behaves as a traditional autoencoder. Previously, it was mentioned how the level of abstraction plays an important role in the quality of image segmentation. To consider various levels of abstraction, U-Net implements skip connections to copy the uncompressed activations from encoding blocks to their mirrored counterparts among the decoding blocks, as shown in the Figure 13. The feature extractor of U-Net can also be upgraded to provide better segmentation maps. The network nicknamed "The 100-layer Tiramisu" [70] applied the concept of U-Net using a dense-net–based feature extractor. Other modern variations involve the use of capsule networks [151] along with locally constrained routing [85]. U-Net was selected as a winner for an ISBI cell tracking challenge. In the PhC-U373 dataset, it scored a mean IoU of 0.9203, whereas the second best was at 0.83. In the DIC-HeLa dataset, it scored a mean IoU of 0.7756, which was significantly better than the second best approach, which scored only 0.46.

4.2.2 Forwarding Pooling Indices. Max pooling has been the most commonly used technique for reducing the size of the activation maps for various reasons. The activations represent the response of the region of an image to a specific kernel. In max pooling, a region of pixels is compressed to a single value by considering only the maximum response obtained within that region. If a typical autoencoder compresses a 2×2 neighborhood of pixels to a single pixel in the encoding phase, the decoder must decompress the pixel to a similar dimension of 2×2. By forwarding pooling indices, the network basically remembers the location of the maximum value among the 4 pixels while performing max pooling. The index corresponding to the maximum value is forwarded to the

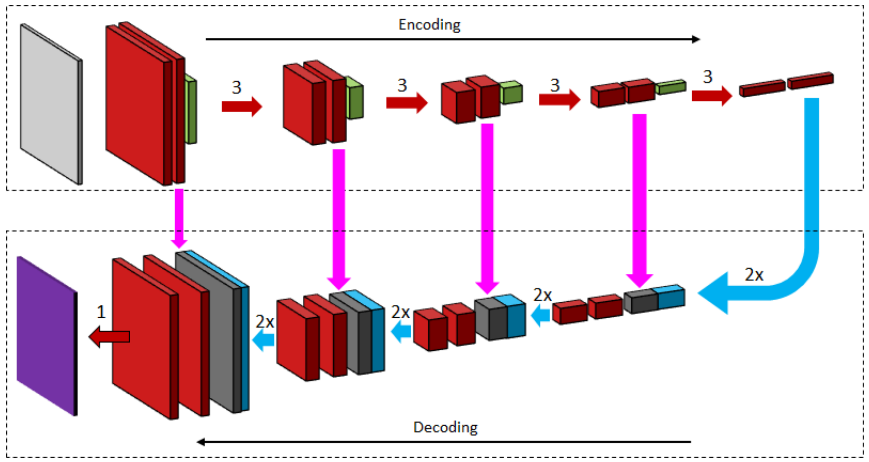

Fig. 13. Architecture of U-Net.

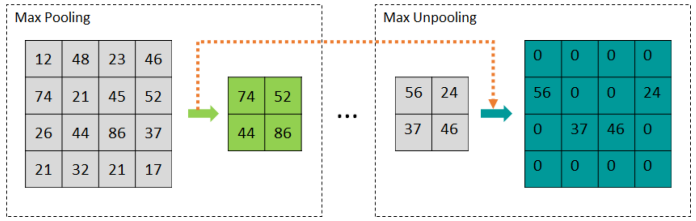

Fig. 14. Forwarding pooling indices to maintain a spatial relationship during unpooling.

decoder (Figure 14) so that during the unpooling operation the value from the single pixel can be copied to the corresponding location in a 2×2 region in the next layer [181]. The values in the rest of the three positions are computed in the subsequent convolutional layers. If the value was copied to a random location without knowledge of the pooling indices, there would be inconsistencies in classification, especially in the boundary regions.

SegNet. The SegNet algorithm [8] was launched in 2015 to compete with the FCN network on complex indoor and outdoor images. The architecture was composed of five encoding blocks and five decoding blocks. The encoding blocks followed the architecture of the feature extractor in the VGG-16 network. Each block is a sequence of multiple convolution, batch normalization, and ReLU layers. Each encoding block ends with a max pooling layer where the indices are stored. Each decoding block begins with a unpooling layer where the saved pooling indices are used (Figure 15). The indices from the max pooling layer of the ith block in the encoder is forwarded to the max unpooling layer in the $(L - i + 1)th$ block in the decoder, where L is the total number of blocks in each encoder and decoder. The SegNet architecture scored a mean IoU of 60.10 compared to 53.88 by DeepLab-Large-FOV [24] or 49.83 by FCN [105] or 59.77 by Deconvnet [127] on the CamVid dataset.

4.3 Adversarial Models

Until now, we have seen purely discriminative models like FCN, DeepMask, and DeepLab that primarily generate a probability distribution for every pixel across the number of classes. Furthermore, in autoencoder-treated segmentation as a generative process, the last layer is generally

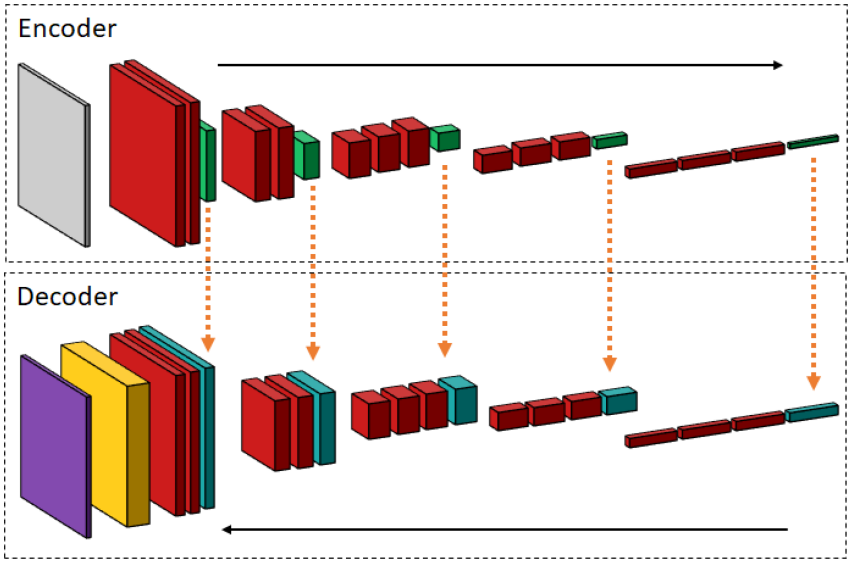

Fig. 15. Architecture of SegNet.

connected to a pixelwise soft-max classifier. The adversarial learning framework approaches the optimization problem from a different perspective. Generative Adversarial Networks (GANs) gained a lot of popularity due to their remarkable performance as a generative network. The adversarial learning framework mainly consists of two networks: a generative network and a discriminator network. The generator G tries to generate images like the ones from the training dataset using a noisy input prior distribution called $p_z(z)$. The network $G(z;\theta_g)$ represents a differentiable function represented by a neural network with weights θ_g. A discriminator network tries to correctly guess whether an input data is from the training data distribution ($p_{data}(x)$) or generated by the generator G. The goal of the discriminator is to get better at catching a fake image, whereas the generator tries to get better at fooling the discriminator, thus in the process generating better outputs. The entire optimization process can be written as a min-max problem as follows:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3}$$

The segmentation problem has also been approached from an adversarial learning perspective. The segmentation network is treated as a generator that generates the segmentation masks for each class, whereas a discriminator network tries to predict whether a set of masks is from the ground truth or from the output of the generator [108]. A schematic diagram of the process is shown in Figure 16. Furthermore, conditional GANs have been used to perform image-to-image translation [68]. This framework can be used for image segmentation problems where the semantic boundaries of the image and output segmentation map do not necessarily coincide, such as in the case of creating a schematic diagram of a façade of a building.

4.4 Sequential Models

Until now, almost all of the techniques discussed deal with semantic image segmentation. Another class of segmentation problem, namely instance-level segmentation, needs a slightly different approach. Unlike semantic image segmentation, here all instances of the same object are segmented into different classes. This type of segmentation problem is mostly handled by learning to give a sequence of object segments as outputs. Hence, sequential models come into play in such problems.

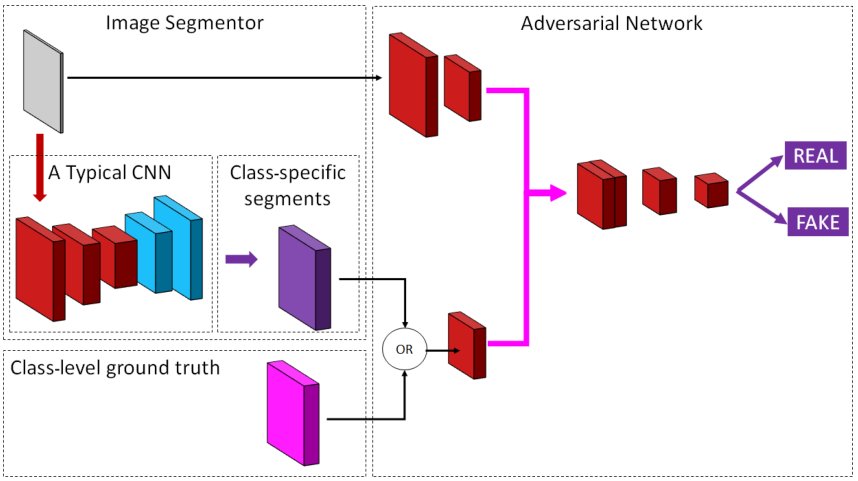

Fig. 16. Adversarial learning model for image segmentation.

Main architectures commonly used are convolutional LSTMs, recurrent networks, attention-based models, and so on.

4.4.1 Recurrent Models. Traditional LSTM networks employ fully connected weights to model long- and short-term memories across sequential inputs. But they fail to capture spatial information of images. Moreover, fully connected weights for images increase the cost of computation greatly. In a convolutional LSTM [146], these weights are replaced by convolutional layers (Figure ??). Convolutional LSTMs have been used in several works to perform instance-level segmentation. Normally, they are used as a suffix to a object segmentation network. The purpose of a recurrent model like LSTM is to select each instance of the object in different timestamps of the sequential output. The approach has been implemented with object segmentation frameworks like FCN and U-Net [21].

4.4.2 Attention Models. Whereas convolutional LSTMs can select different instance of objects at different timestamps, attention models are designed to have more control over this process of localizing individual instances. One simple method to control attention is by spatial inhibition [146]. Spatial inhibition network is designed to learn a bias parameter that cuts off previously detected segments from future activations. Attention models have been further developed with the introduction of a dedicated attention module and an external memory to keep track of segments. In the work of Ren and Zemel [144], the instance segmentation network was divided into four modules. First, an external memory provides object boundary details from all previous steps. Second, a box network attempts to predict the location of the next instance of the object and outputs a sub-region of the image for the third module, which is the segmentation module. The segmentation module is similar to the convolutional autoencoder model discussed previously. The fourth module scores the predicted segments based on whether they qualify as a proper instance of the object. The network terminates when the score goes below a user-defined threshold.

4.5 Weakly Supervised or Unsupervised Models

Neural networks in general are trained with algorithms like back-propagation, where the parameters $\mathbf{w}$ are updated based on their local partial derivative with respect to an error value E obtained

using a loss function f.

$$w = w + \Delta w = w - \eta \frac{\delta E}{\delta w} \tag{4}$$

The loss function is generally expressed in terms of a distance between a target value and the predicted value. But in many scenarios, image segmentation requires the use of data without annotations with ground truth. This leads to the development of unsupervised image segmentation techniques. One of the straightforward ways to achieve this is to use networks pre-trained on other larger datasets with similar kinds of samples and ground truths and use clustering algorithms like k-means on the feature maps. However, this kind of semi-supervised technique is inefficient for data samples that have a unique distribution of sample space. Another con is that the network is trained to perform on a input distribution that is still different from the test data. That does not allow the network to perform to its full potential. The key problem in fully unsupervised segmentation algorithm is the development of a loss function capable of measuring the quality of segments or cluster of pixels. With all of these limitations, the amount of literature is comparatively much lighter when it comes to weakly supervised or unsupervised approaches.

4.5.1 Weakly Supervised Algorithms. Even in the lack of proper pixel-level annotations, segmentation algorithms can exploit coarser annotations like bounding boxes or even image-level labels [92, 132] for performing pixel-level segmentation.

Exploiting Bounding Boxes. From the angle of data annotation, defining bounding boxes is a much less expensive task compared to pixel-level segmentation. The availability of datasets with bounding boxes is also much larger than those with pixel-level segmentations. The bounding box can be used as a weak supervision to generate pixel-level segmentation maps. In the work of Dai et al. [34], titled "BoxSup," segmentation proposals were generated using region proposal methods like selective search. After that, multi-scale combinatorial grouping was used to combine candidate masks with an objective to select the optimal combination that has the highest IOU with the box. This segmentation map was used to tune a traditional image segmentation network like FCN. BoxSup was able to attain a mean IOU of 75.1 in the Pascal VOC 2012 test set compared to 62.2 of FCN or 66.4 of DeepLab-CRF.

4.5.2 Unsupervised Segmentation. Unlike supervised or weakly supervised segmentation, the success of unsupervised image segmentation algorithms are mostly dependent on the learning mechanism. Some of the common approaches are described next.

Learning Multiple of Objectives. One of the most generic approaches in unsupervised learning is considering multiple objectives designed in a way to perform well for segmentation when ground truths are not available.

A common variant called *JULE* or joint unsupervised learning of deep representation have been used in several applications where there is a lack of samples with ground truth. The basis of JULE lies in training a sequential model along with a deep feature extraction model. The learning methodology was primarily introduced as an image clustering algorithm. However, it has been extended to other applications like image segmentation. The key objective for being able to perform these kinds of segmentation is the development of a proper objective function. In JULE, the objective function considers the affinity between samples in the clusters and also the negative affinity between the clusters and its neighbors. Agglomerative clustering is performed across the timestamps of a recurrent network. In the work of Moriya et al. [122], JULE was attempted on image patches rather than entire samples of images. With a similar objective function, it was able to classify the patches into a predefined number of classes. JULE was used in this case to provide the agglomerative clustering information as supervisory signals for the next iteration.

Another variant of learning multiple objectives has been demonstrated through adversarial collaboration [142] among neural networks with independent jobs like monocular depth prediction, estimating camera motion, detecting optical flow, and segmentation of a video into the static scene and moving regions. Through competitive collaboration, each network competes to explain the same pixel that either belongs to a static or moving class, which in turn shares the learned concepts with a moderator to perform the motion segmentation.

Using Refinement Modules for Self-Supervision. Using other unsupervised over-segmentation techniques can be used to provide supervision to deep feature extractors [74] and by enforcing multiple constraints like similarity between features, spatial continuity and intra-axis normalization. All of these objectives are optimized through back-propagation. The spatial continuity is achieved by extracting a superpixel from the image using standard algorithms like the SLIC [1], and all pixels within a superpixel are forced to have the same label. The difference between the two segmentation maps is used as a supervisory signal to update the weights.

Other Relevant Unsupervised Techniques for Extracting Semantic Information. Learning without annotations is always challenging, and hence there is a variety of literature in which many interesting solutions have been proposed. Using CNNs to solve jigsaw puzzles [128] derived from images can be used to learn semantic connections between various parts of the objects. The proposed context-free network takes a set of image tiles as input and tries to establish the correct spatial relations between them. During the process, it simultaneously learns features specific to parts of an object, as well as their semantic relationships. These patch-based self-supervision techniques can be further improved using contextual information [123]. Using context encoders [134] can also derive a spatial and semantic relationship among various parts of an image. Context encoders basically are CNNs trained to generate arbitrary regions of an image that is conditioned by its surrounding information. Another demonstration of extracting semantic information can be found in the process of image colorization [184]. The process of colorization requires pixel-level understanding of the semantic boundaries corresponding to objects. Other self-supervision techniques can leverage motion cues in videos to segment various parts of an object for better semantic understanding [182]. This method can learn several structural and coherent features for tasks like semantic segmentation, human parsing, and instance segmentation.

4.5.3 W-Net. W-Net [173] derived its inspiration from the previously discussed U-Net. The W-Net architecture consists of two cascaded U-Nets. The first U-Net acts as a encoder that converts an image to its segmented version, whereas the second U-Net tries to reconstruct the original image from the output of the first U-Net that is the segmented image. Two loss functions are minimized simultaneously, one of them being the mean square error between the input image and the reconstructed image that is given by the second U-Net. The second loss function is derived from the normalized cut [154]. The hard normalized cut is formulated as

$$Ncut_K(V) = \sum_{k=1}^{K} \frac{\sum_{u \in A_k, v \in V - A_k} w(u, v)}{\sum_{u \in A_k, t \in V} w(u, t)}, \tag{5}$$

where A_k is set of pixels in the kth segment, V is the set of all pixels, and w measures the weight between two pixels.

However, this function is not differentiable, and hence back-propagation is not possible. Therefore, a soft version of function was proposed:

$$J_{soft-Ncut}(V, K) = K - \sum_{k=1}^{K} \frac{\sum_{u \in V} p(u = A_k) \sum_{v \in V} w(u, v) p(v = A_k)}{\sum_{u \in V} p(u = A_k) \sum_{t \in V} w(u, t)}, \tag{6}$$

where $p(u = A_k)$ represents the probability of a node u belonging to a class A_k. The output segmentation maps were further refined using fully connected CRFs. Remaining insignificant segments were further merged using hierarchical clustering.

4.6 Interactive Segmentation

Image segmentation is one of the most difficult challenges in the field of computer vision. In many scenarios where the images are too complex, noisy, or subject to poor illumination conditions, a little bit of interaction and guidance from users can significantly improve the performance of segmentation algorithms. Interactive segmentation has been flourishing even outside the deep learning paradigm. However, with powerful feature extraction of CNNs, the amount of interaction can be reduced to a great extent.

4.6.1 Two Stream Fusion. One of the most straightforward implementations of interactive segmentation is to have two parallel branches, one from the image and another from an image representing the interactive stream, and fuse them to perform the segmentation [65]. The interaction inputs are taken in the form of different colored dots representing positive and negative classes. With a bit of post-processing where intensities of the interaction maps are calculated based on the euclidean distance from the points, we get two sets of maps (one for each class) that look like fuzzy Voronoi cells with the points at the center. The maps are multiplied elementwise to obtain the image for the interaction stream. The two branches are composed of a sequence of convolutions and pooling layers. The Hadamard product of features obtained at the end of each branch are sent to the fusion network where a low-resolution segmentation map is generated. An alternative approach follows the footsteps of FCN to fuse features of multiple scales to obtain a sharper-resolution image.

4.6.2 Deep Extreme Cut. Contrary to the two stream approach, deep extreme cut [112] takes a single pipeline to create segmentation maps from RGB images. This method expects four points from the user denoting the four extreme regions in the boundary of the object (left most, right most, top most, bottom most). By creating heatmaps from the points, a four-channel input is fed into a DenseNet-101 network. The final feature map of the network is passed into a pyramid scene parsing module for analyzing global contexts to perform the final segmentation. This method was able to attain a mean IOU of 80.3 on the PASCAL test set.

4.6.3 Polygon-RNN. Polygon-RNN [19] takes a different approach than the other methods. Multi-scale features are extracted from different layers of a typical VGG network and concatenated to create a feature block for a recurrent network. The RNN in turn is supposed to provide a sequence of points as an output that represents the contour of the object. The system is primarily designed as an interactive image annotation tool. The users can interact in two different ways. First, the users must provide a tight bounding box for the object of interest. Second, after the polygon is built, the users are allowed to edit any point in the polygon. However, this editing is not used for any further training of the system and hence presents a small avenue for improvement of the system.

4.7 Building More Efficient Networks

Although many complicated networks with lots of fancy modules can give a very decent quality of semantic segmentation, embedding such algorithms in real-life systems is a different story. Many other factors, such as the cost of hardware and real-time response, pose a new degree of challenge. Efficiency is also key for creating consumer-level systems.

4.7.1 ENet. ENet [133] brought forward a couple of interesting design choices to create a quite shallow segmentation network with a small number of parameters (0.37 million). Instead of a symmetric encoder-decoder architecture like SegNet or U-Net, it has a deeper encoder and a shallower decoder. Instead of increasing channel sizes after pooling, parallel pooling operations were performed along with convolutions of stride 2 to reduce overall features. To increase the learning capability, PReLU was used compared to ReLU so that the transfer functions remain dynamic and it can simulate the jobs of ReLU and an identity function as required. This is normally an important factor in ResNet; however, because the network is shallow, using PReLU is a smarter choice. Using factorized filters also allows a smaller number of parameters.

4.7.2 Deep Layer Cascade. Deep Layer Cascade [92] tackles several challenges and makes two significant contributions. First, it analyzes the level of difficulty of pixel-level segmentation for various classes. With a cascaded network, easier segments are discovered in the earlier stage, whereas the latter layers focus on regions that need more delicate segments. Second, the proposed layer cascading can be used with common networks like Inception-ResNet-V2 (IRNet) to improve the speed and even the performance to some extent. The basic principle of IRNet is to create a multistage pipeline where in each stage a certain amount of pixels would be classified into one of the segments. In the earlier stages, the easiest pixels will be classified and the harder pixels with more uncertainty will move forward to latter stages. In the subsequent stages, the convolutions will only take place on those pixels that could not be classified in the previous stage while forwarding yet harder pixels to the next stage. Typically, the proposed model comes with three stages, each adding more convolutional modules to the network. With layer cascading, a mean IOU of 82.7 was reached on the VOC12 test set, with DeepLab-V2 and Deep Parsing Network being the nearest competitors with a mean IOU of 79.7 and 77.5, respectively. In terms of speed, 23.6 frames per second (fps) were processed compared to 14.6fps by SegNet or 7.1fps by DeepLab-V2.

4.7.3 SegFast. Another recent implementation titled "SegFast" [130] was able to build a network with only 0.6 million parameters that resulted in a network that can do a forward pass in around 0.38 seconds without a GPU. The approach combined the concept of depthwise separable convolutions with the fire modules of SqueezeNet. SqueezeNet introduced the concepts of fire modules to reduce the number of convolutional weights. With depthwise separable convolutions, the number of parameters went further down. They also proposed the use of depthwise separable transposed convolutions for decoding. Even with so many approaches of feature reductions, the performance was quite comparable to other popular networks, such as SegNet.

4.7.4 Segmentation Using Super-Pixels. Over-segmentation algorithms [1] have flourished well to divide images into small patches based on local information. With patch classification algorithms, these super-pixels can be converted to semantic segments. However, since the process of over-segmentation does not consider neighborhood relations, it is necessary to include that in the patch classification algorithm. It is much faster to perform patch classification compared to pixel-level classification simply because of the fact that the number of super-pixels is much less than the number of pixels in an image. One of the first works on using CNNs for super-pixel–level classification was carried out by Farabet et al. [45]. However, just considering super-pixels without context can result in erroneous classification. In the work of Das et al. [37], multiple levels of contexts were captured by considering neighborhood super-pixels of different levels during the patch classification. By fusing patch-level probabilities from different levels of contexts using methods like weighted averaging, max-voting, or uncertainty-based approaches like the Dempster-Shafer theory, a very efficient segmentation algorithm was proposed. Das et al. [37] were able to obtain a pixel-level accuracy of 77.14% as opposed to 74.56% by Farabet et al. [45]. Although

super-pixels can be exploited to build efficient semantic segmentation models, the reverse is also true. Conversely, semantic segmentation ground truths can be used to train networks to perform over-segmentation [164]. Pixel affinities can be calculated using convolutional features to perform over-segmentation while paying special attention to semantic boundaries.

5 APPLICATIONS

Image segmentation is one of the most commonly addressed problems in the domain of computer vision. It is often augmented with other related tasks like object detection, object recognition, scene parsing, image description generation. Hence this branch of study finds extensive use in various real-life scenarios.

5.1 Content-Based Image Retrieval

With the ever-increasing amount of structured and unstructured data on the Internet, development of efficient information retrieval systems is of utmost importance. Content-Based Image Retrieval (CBIR) systems have hence been a lucrative area of research. Similar interests also exist in many other related problems like visual question answering, interactive query–based image processing, description generation. Image segmentation is useful in many cases, as it is representative of spatial relations among various objects [10, 102]. Instance-level segmentation is essential for handling numeric queries [183]. Unsupervised approaches [72] are particularly useful for handling a bulk amount of non-annotated data, which is very common in this field of work.

5.2 Medical Imaging

Another major application area for image segmentation is in the domain of health care. Many kinds of diagnostic procedures involve working with images corresponding to different types of imaging sources and various parts of the body. Some of the most common types of tasks are segmentation of organic elements such as vessels [47], tissues [73], and nerves [107]. Other kinds of problems include localization of abnormalities like tumors [118, 181] and aneurysms [39, 106]. Microscopic images [67] also need various kinds of segmentations, such as cell or nuclei detection, counting numbers of cells, and cell structure analysis for cancer detection. The primary challenges with this domain is the lack of a bulk amount of data for challenging diseases and variety in the quality of images due to the different types of imaging devices involved. Medical procedures not only involve human beings but also animals and plants.

5.3 Object Detection

With the success of deep learning algorithms, there has also been a surge in research areas related to automatic object detection, such as robotic maneuverability [90], autonomous driving [163], intelligent motion detection [160], and tracking systems [170]. Extremely remote regions, such as the deep sea [83, 158] or space [149], can efficiently be explored with the help of intelligent robots making autonomous decisions. In sectors like defense, unmanned aerial vehicles (UAVs) [125] are used to detect anomalies or threats in remote regions [95]. Segmentation algorithms have significant use in satellite images for various geo-statistical analysis [86]. In fields like image or video post-production, it is often essential to perform segmentation for various tasks, such as image matting [91], compositing [17], and rotoscoping [2].

5.4 Forensics

Biometric verification systems that use the iris [52, 100], fingerprints [76], finger veins [140], and dental records [75] involve segmentation of various informative regions for efficient analysis.

5.5 Surveillance

Surveillance systems [71, 77, 120] are associated with various issues, such as occlusion, lighting, or weather conditions. Moreover, surveillance systems can also involve analysis of images from hyper-spectral sources [4]. Surveillance systems can also be extended to applications such as object tracking [64], searching [3], anomaly detection [143], threat detection [111], and traffic control [174]. Image segmentation plays a vital role in segregating objects of interest from the clutter present in natural scenes.

6 DISCUSSION AND FUTURE SCOPE

Throughout this article, various methods have been discussed in an effort to highlight their key contributions, pros, and cons. With so many different options, it is still hard to choose the right approach to a problem. The most optimal way to choose a correct algorithm is to first analyze the variables that affect the choice.

One of the most important aspects that affect the performance of deep learning–based approaches is the availability of datasets and annotations. In that regard, a concise list of datasets belonging to various domains was provided in Table 1. When working on other small-scale datasets, it is a common practice to pre-train the network on a larger dataset of a similar domain. Sometimes an ample amount of samples are available, yet pixel-level segmentation labels may not be available considering that creating them is a taxing problem. Even in those cases, pre-training parts of networks on other related problems like classification or localization can also help in the process of learning a better set of weights.

A related decision that one must make in this regard is to choose among supervised, unsupervised, or weakly supervised algorithms. In the current scenario, a large number of supervised approaches exist; however, unsupervised and weakly supervised algorithms are still far from reaching a level of saturation. This is a legitimate concern in the field of image segmentation because data collection can be carried out through many automated processes, but annotating them perfectly requires manual labor. It is one of the most prominent areas where a researcher can contribute in terms of building end-to-end scalable systems that can model data distribution, decide on the optimal number of classes, and create accurate pixel-level segmentation maps in a completely unsupervised domain. The area of weakly supervised algorithms is also highly demanding. It is much easier to collect annotations corresponding to problems like classification or localization. Using those annotations to guide the image segmentation problem is also a promising domain.

The next important aspect of building deep learning models for image segmentation is the selection of the appropriate approaches. Pre-trained classifiers can be used for various fully convolutional approaches. Most of the time, some kind of multi-scale feature fusion can be carried out by combining information from different depths of the network. Pre-trained classifiers like VGGNet or ResNet or DenseNet are also often used for the encoder part of a encoder-decoder architecture. Here also, information can be passed from various layers of encoders to corresponding similar-size layers of the decoder to obtain multi-scale information. Another major benefit of encoder-decoder architectures is that if the down-sampling and up-sampling operations are designed carefully, outputs can be generated that are of the same size as the input. It is a major benefit over simple convolutional approaches like FCN or DeepMask. This removes the dependency on the input size and hence makes the system more scalable. These two approaches are the most common in case of a semantic segmentation problem. However, if a finer level of instance-specific segments are required, it is often necessary to couple with other methods corresponding to object detection. Utilizing bounding box information is one way to address these problems, whereas

other approaches use attention-based models or recurrent models to provide output as sequence of segments for each instance of the object.

There can be two aspects to consider while measuring the performance of the system. One is speed, and the other is accuracy. CRF is one of the most commonly used post-processing modules for refining outputs from other networks. CRFs can be simulated as an RNN to create end-to-end trainable modules to provide very precise segmentation maps. Other refinement strategies include the use of over-segmentation algorithms like super-pixels, or using human interactions to guide segmentation algorithms. In terms of gain in speed, networks can be highly compressed using strategies like depthwise separable convolutions, kernel factorizations, and reducing the number of spatial convolutions. These tactics can reduce the number of parameters to a great extent without reducing the performance too much. Lately, generative adversarial networks have seen a tremendous rise in popularity. However, their use in the field of segmentation is still pretty thin, with only a handful of approaches addressing the avenue. Given the success that they have gained, they certainly have the potential to improve existing systems by a great margin.

The future of image segmentation largely depends on the quality and quantity of available data. Although there is an abundance of unstructured data on the Internet, the lack of accurate annotations is a legitimate concern. Pixel-level annotations especially can be incredibly difficult to obtain without manual intervention. The most ideal scenario would be to exploit the data distribution itself to analyze and extract meaningful segments that represent concepts rather than content. This is an incredibly challenging task especially if we are working with a huge amount of unstructured data. The key is to map a representation of the data distribution to the intent of the problem statement such that the derived segments are meaningful in some way and contribute to the overall purpose of the system.

7 CONCLUSION

Image segmentation has seen a new rush of deep learning–based algorithms. Starting with the evolution of deep learning–based algorithms, we have thoroughly explained the pros and cons of the various state-of-the-art algorithms associated with image segmentation based on deep learning. The simple explanations allow the reader to grasp the most basic concepts that contribute to the success of deep learning–based image segmentation algorithms. The unified representation scheme followed in the figures can highlight the similarities and differences of various algorithms. In the future, this theoretical survey work can be accompanied by empirical analysis of the discussed methods.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable suggestions which helped improve the quality of the article.

REFERENCES

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, et al. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, 11 (2012), 2274–2282.

[2] Aseem Agarwala, Aaron Hertzmann, David H. Salesin, and Steven M. Seitz. 2004. Keyframe-based tracking for rotoscoping and animation. 23, 584–591.

[3] Jamil Ahmad, Irfan Mehmood, and Sung Wook Baik. 2017. Efficient object-based surveillance image search using spatial pooling of convolutional features. *Journal of Visual Communication and Image Representation* 45 (2017), 62–76.

[4] Fahim Irfan Alam, Jun Zhou, Alan Wee-Chung Liew, and Xiuping Jia. 2016. CRF learning with CNN features for hyperspectral image segmentation. In *Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS'16)*. IEEE, Los Alamitos, CA, 6890–6893.

[5] Alberto Albiol, Luis Torres, and Edward J. Delp. 2001. An unsupervised color image segmentation algorithm for face detection applications. In *Proceedings of the 2001 International Conference on Image Processing*, Vol. 2. IEEE, Los Alamitos, CA, 681–684.

[6] Teresa Araújo, Guilherme Aresta, Eduardo Castro, José Rouco, Paulo Aguiar, Catarina Eloy, António Polónia, and Aurélio Campilho. 2017. Classification of breast cancer histology images using convolutional neural networks. *PloS One* 12, 6 (2017), e0177544.

[7] Aamer Ather. 2009. *A Quality Analysis of OpenStreetMap Data*. Master's Thesis. University College London, London, UK.

[8] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (2017), 2481–2495.

[9] John Barlow, Steven Franklin, and Yvonne Martin. 2006. High spatial resolution satellite imagery, DEM derivatives, and image segmentation for the detection of mass wasting processes. *Photogrammetric Engineering and Remote Sensing* 72, 6 (2006), 687–692.

[10] Serge Belongie, Chad Carson, Hayit Greenspan, and Jitendra Malik. 1998. Color-and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the 6th International Conference on Computer Vision*. IEEE, Los Alamitos, CA, 675–682.

[11] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*. 153–160.

[12] L. Sant'Anna Bins, L. M. Garcia Fonseca, G. J. Erthal, and F. Mitsuo Ii. 1996. Satellite imagery segmentation: A region growing approach. *Simpósio Brasileiro de Sensoriamento Remoto* 8, 1996 (1996), 677–680.

[13] Ali Borji. 2015. What is a salient object? A dataset and a baseline model for salient object detection. *IEEE Transactions on Image Processing* 24, 2 (2015), 742–756.

[14] Ali Borji, Ming-Ming Cheng, Qibin Hou, Huaizu Jiang, and Jia Li. 2014. Salient object detection: A survey. arXiv:1411.5878.

[15] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. 2015. Salient object detection: A benchmark. *IEEE Transactions on Image Processing* 24, 12 (2015), 5706–5722.

[16] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. 2009. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters* 30, 2 (2009), 88–97.

[17] Nathan D. Cahill and Lawrence A. Ray. 2007. Method and system for compositing images to produce a cropped image. US Patent 7,162,102.

[18] Aaron Carass, Snehashis Roy, Amod Jog, Jennifer L. Cuzzocreo, Elizabeth Magrath, Adrian Gherman, et al. 2017. Longitudinal multiple sclerosis lesion segmentation data resource. *Data in Brief* 12 (2017), 346–350.

[19] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. 2017. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5230–5238.

[20] Ping-Lin Chang and Wei-Guang Teng. 2007. Exploiting the self-organizing map for medical image segmentation. In *Proceedings of the 20th IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)*. IEEE, Los Alamitos, CA, 281–288.

[21] Jianxu Chen, Lin Yang, Yizhe Zhang, Mark Alber, and Danny Z. Chen. 2016. Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation. In *Advances in Neural Information Processing Systems*. 3036–3044.

[22] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2014. Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv:1412.7062.

[23] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2018. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 4 (2018), 834–848.

[24] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1529–1537.

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

[26] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. 2016. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3640–3649.

[27] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. arXiv:1802.02611.

[28] Kuo-Sheng Cheng, Jzau-Sheng Lin, and Chi-Wu Mao. 1996. The application of competitive Hopfield neural network to medical image segmentation. *IEEE Transactions on Medical Imaging* 15, 4 (1996), 560–567.

[29] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, and Shi-Min Hu. 2014. SalientShape: Group saliency in image collections. *Visual Computer* 30, 4 (2014), 443–453.
[30] Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu. 2015. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 3 (2015), 569–582. https://mmcheng.net/msra10k/.
[31] Keh-Shih Chuang, Hong-Long Tzeng, Sharon Chen, Jay Wu, and Tzong-Jer Chen. 2006. Fuzzy c-means clustering with spatial information for image segmentation. *Computerized Medical Imaging and Graphics* 30, 1 (2006), 9–15.
[32] Dorin Comaniciu and Peter Meer. 1997. Robust analysis of feature spaces: Color image segmentation. In *Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, Los Alamitos, CA, 750–755.
[33] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3213–3223.
[34] Jifeng Dai, Kaiming He, and Jian Sun. 2015. BoxSup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1635–1643.
[35] Jifeng Dai, Kaiming He, and Jian Sun. 2016. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3150–3158.
[36] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. 2016. R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*. 379–387.
[37] Aritra Das, Swarnendu Ghosh, Ritesh Sarkhel, Sandipan Choudhuri, Nibaran Das, and Mita Nasipuri. 2019. Combining multilevel contexts of superpixel using convolutional neural networks to perform natural scene labeling. In *Recent Developments in Machine Learning and Data Analytics*. Springer, 297–306.
[38] M. Portes De Albuquerque, I. A. Esquef, and A. R. Gesualdi Mello. 2004. Image thresholding using Tsallis entropy. *Pattern Recognition Letters* 25, 9 (2004), 1059–1065.
[39] Marleen de Bruijne, Bram van Ginneken, Max A. Viergever, and Wiro J. Niessen. 2004. Interactive segmentation of abdominal aortic aneurysms in CTA images. *Medical Image Analysis* 8, 2 (2004), 127–138.
[40] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, Devis Tuia, and Ramesh Raskar. 2018. DeepGlobe 2018: A challenge to parse the earth through satellite images. arXiv:1805.06561.
[41] Yingzi Du, Emrah Arslanturk, Zhi Zhou, and Craig Belcher. 2011. Video-based noncooperative iris image segmentation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41, 1 (2011), 64–74.
[42] Lixin Duan, Ivor W. Tsang, Dong Xu, and Tat-Seng Chua. 2009. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, New York, NY, 289–296.
[43] Vincent Dumoulin and Francesco Visin. 2016. A guide to convolution arithmetic for deep learning. arXiv:1603.07285.
[44] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88, 2 (2010), 303–338.
[45] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. 2013. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1915–1929.
[46] International Society for Photogrammetry and Remote Sensing. [n.d.]. ISPRS 2D Semantic Labeling Contest. Retrieved August 1, 2019 from http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html
[47] Muhammad Moazam Fraz, Paolo Remagnino, Andreas Hoppe, Bunyarit Uyyanonvara, Alicja R. Rudnicka, Christopher G. Owen, and Sarah A. Barman. 2012. Blood vessel segmentation methodologies in retinal images—A survey. *Computer Methods and Programs in Biomedicine* 108, 1 (2012), 407–433.
[48] Jordi Freixenet, Xavier Muñoz, David Raba, Joan Martí, and Xavier Cufí. 2002. Yet another survey on image segmentation: Region and boundary information integration. In *Proceedings of the European Conference on Computer Vision*. 408–422.
[49] Nir Friedman and Stuart Russell. 1997. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence*. 175–181.
[50] K.-S. Fu and J. K. Mui. 1981. A survey on image segmentation. *Pattern Recognition* 13, 1 (1981), 3–16.
[51] Fabio Galasso, Naveen Shankar Nagaraja, Tatiana Jimenez Cardenas, Thomas Brox, and Bernt Schiele. 2013. A unified video segmentation benchmark: Annotation, metrics and analysis. In *Proceedings of the IEEE International Conference on Computer Vision*. 3527–3534.
[52] Abhishek Gangwar and Akanksha Joshi. 2016. DeepIrisNet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition. In *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP'16)*. IEEE, Los Alamitos, CA, 2301–2305.
[53] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. 2017. A review on deep learning techniques applied to semantic segmentation. arXiv:1704.06857.

[54] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The Kitti vision benchmark suite. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*. IEEE, Los Alamitos, CA, 3354–3361.

[55] Qichuan Geng, Zhong Zhou, and Xiaochun Cao. 2018. Survey of recent progress in semantic image segmentation with CNNs. *Science China Information Sciences* 61, 5 (2018), 051101.

[56] Ross Girshick. 2015. Fast R-CNN. arXiv:1504.08083.

[57] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 580–587.

[58] Stephen Gould, Richard Fulton, and Daphne Koller. 2009. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*. IEEE, Los Alamitos, CA, 1–8.

[59] Xiao Han. 2017. Automatic liver lesion segmentation using a deep convolutional neural network method. arXiv:1704.07239. https://competitions.codalab.org/competitions/17094.

[60] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. 2011. Semantic contours from inverse detectors. In *Proceedings of the International Conference on Computer Vision (ICCV'11)*.

[61] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV'17)*. IEEE, Los Alamitos, CA, 2980–2988.

[62] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 9 (2015), 1904–1916.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[64] Seunghoon Hong, Tackgeun You, Suha Kwak, and Bohyung Han. 2015. Online tracking by learning discriminative saliency map with convolutional neural network. In *Proceedings of the International Conference on Machine Learning*. 597–606.

[65] Yang Hu, Andrea Soltoggio, Russell Lock, and Steve Carter. 2019. A fully convolutional two-stream fusion network for interactive image segmentation. *Neural Networks* 109 (2019), 31–42.

[66] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

[67] Humayun Irshad, Antoine Veillard, Ludovic Roux, and Daniel Racoceanu. 2014. Methods for nuclei detection, segmentation, and classification in digital histopathology: A review—Current status and future potential. *IEEE Reviews in Biomedical Engineering* 7 (2014), 97–114.

[68] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-image translation with conditional adversarial networks. arXiv:1611.07004.

[69] Firas Ajil Jassim and Fawzi H. Altaani. 2013. Hybridization of Otsu method and median filter for color image segmentation. arXiv:1305.1052.

[70] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. 2017. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'17)*. IEEE, Los Alamitos, CA, 1175–1183.

[71] Cheng-Bin Jin, Shengzhe Li, Trung Dung Do, and Hakil Kim. 2015. Real-time human action recognition using CNN over temporal images for static video surveillance cameras. In *Proceedings of the Pacific Rim Conference on Multimedia*. 330–339.

[72] A. H. Kam, T. T. Ng, N. G. Kingsbury, and W. J. Fitzgerald. 2000. Content based image retrieval through object extraction and querying. In *Proceedings of the Workshop on Content-Based Access of Image and Visual Libraries (CBAIVL'00)*. IEEE, Los Alamitos, CA, 91.

[73] Konstantinos Kamnitsas, Christian Ledig, Virginia F. J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker. 2017. Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *Medical Image Analysis* 36 (2017), 61–78.

[74] Asako Kanezaki. 2018. Unsupervised image segmentation by backpropagation. In *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18)*. IEEE, Los Alamitos, CA, 1543–1547.

[75] Jiayin Kang, Xiao Li, Qingxian Luan, Jinzhu Liu, and Lequan Min. 2006. Dental plaque quantification using cellular neural network-based image segmentation. In *Intelligent Computing in Signal Processing and Pattern Recognition*. Springer, 797–802.

[76] Jiayin Kang and Wenjuan Zhang. 2009. Fingerprint segmentation using cellular neural network. In *Proceedings of the International Conference on Computational Intelligence and Natural Computing (CINC'09)*, Vol. 2. IEEE, Los Alamitos, CA, 11–14.

[77] Kai Kang and Xiaogang Wang. 2014. Fully convolutional neural networks for crowd segmentation. arXiv:1411.4464.
[78] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.
[79] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. 2016. HyperNet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 845–853.
[80] Philipp Krähenbühl and Vladlen Koltun. 2011. Efficient inference in fully connected CRFs with Gaussian edge potentials. In *Advances in Neural Information Processing Systems*. 109–117.
[81] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2016. Semantic understanding of scenes through the ADE20K dataset. arXiv:1608.05442.
[82] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.
[83] Alfonso B. Labao and Prospero C. Naval. 2017. Weakly-labelled semantic segmentation of fish objects in underwater videos using a deep residual network. In *Proceedings of the Asian Conference on Intelligent Information and Database Systems*. 255–265.
[84] W. Ladys Law Skarbek and Andreas Koschan. 1994. Colour image segmentation a survey. *IEEE Transactions on Circuits and Systems for Video Technology* 14, 7 (1994).
[85] Rodney LaLonde and Ulas Bagci. 2018. Capsules for object segmentation. arXiv:1804.04241.
[86] Martin Längkvist, Andrey Kiselev, Marjan Alirezaie, and Amy Loutfi. 2016. Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sensing* 8, 4 (2016), 329.
[87] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.
[88] Seong-Hun Lee, Min Su Cho, Kyomin Jung, and Jin Hyung Kim. 2010. Scene text extraction with edge constraint and text collinearity. In *Proceedings of the 2010 International Conference on Pattern Recognition*. IEEE, Los Alamitos, CA, 3983–3986.
[89] Bastian Leibe, Edgar Seemann, and Bernt Schiele. 2005. Pedestrian detection in crowded scenes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. IEEE, Los Alamitos, CA, 878–885.
[90] Dan Levi, Noa Garnett, Ethan Fetaya, and Israel Herzlyia. 2015. StixelNet: A deep convolutional network for obstacle detection and road segmentation. In *Proceedings of the British Machine Vision Association (BMVC'15)*. 109.
[91] Anat Levin, Dani Lischinski, and Yair Weiss. 2008. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 2 (2008), 228–242.
[92] Xiaoxiao Li, Ziwei Liu, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2017. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3193–3202.
[93] Yin Li, Xiaodi Hou, Christof Koch, James M. Rehg, and Alan L. Yuille. 2014. The secrets of salient object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 280–287.
[94] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. 2016. Fully convolutional instance-aware semantic segmentation. arXiv:1611.07709.
[95] Wen-Nung Lie. 1995. Automatic target segmentation by locally adaptive image thresholding. *IEEE Transactions on Image Processing* 4, 7 (1995), 1036–1041.
[96] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. 2017. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*.
[97] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. arXiv:1312.4400.
[98] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*. 740–755.
[99] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram van Ginneken, and Clara I Sánchez. 2017. A survey on deep learning in medical image analysis. *Medical Image Analysis* 42 (2017), 60–88.
[100] Nianfeng Liu, Haiqing Li, Man Zhang, Jing Liu, Zhenan Sun, and Tieniu Tan. 2016. Accurate iris segmentation in non-cooperative environments using fully convolutional networks. In *Proceedings of the 2016 International Conference on Biometrics (ICB'16)*. IEEE, Los Alamitos, CA, 1–8.
[101] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. 2017. Learning affinity via spatial propagation networks. In *Advances in Neural Information Processing Systems*. 1520–1530.
[102] Ying Liu, Dengsheng Zhang, Guojun Lu, and Wei-Ying Ma. 2007. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition* 40, 1 (2007), 262–282.

[103] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. 2015. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*. 1377–1385.

[104] Christos P. Loizou, Víctor Murray, Marios S. Pattichis, Ioannis Seimenis, Marios Pantziaris, and Constantinos S. Pattichis. 2011. Multiscale amplitude-modulation frequency-modulation (AM–FM) texture analysis of multiple sclerosis in brain MRI images. *IEEE Transactions on Information Technology in Biomedicine* 15, 1 (2011), 119–129.

[105] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3431–3440.

[106] Karen López-Linares, Nerea Lete, Luis Kabongo, Mario Ceresa, Gregory Maclair, Ainhoa García-Familiar, Iván Macía, and Miguel Ángel González Ballester. 2018. Comparison of regularization techniques for DCNN-based abdominal aortic aneurysm segmentation. In *Proceedings of the 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI'18)*. IEEE, Los Alamitos, CA, 864–867.

[107] Ping Lu, Livia Barazzetti, Vimal Chandran, Kate Gavaghan, Stefan Weber, Nicolas Gerber, and Mauricio Reyes. 2018. Highly accurate facial nerve segmentation refinement from CBCT/CT imaging using a super-resolution classification approach. *IEEE Transactions on Biomedical Engineering* 65, 1 (2018), 178–188.

[108] Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. 2016. Semantic segmentation using adversarial networks. arXiv:1611.08408.

[109] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. 2017. Can semantic labeling methods generalize to any city? The INRIA aerial image labeling benchmark. In *Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing (IGARSS'17)*.

[110] Oskar Maier, Bjoern H. Menze, Janina von der Gablentz, Levin Häni, Mattias P. Heinrich, Matthias Liebrand, et al. 2017. ISLES 2015—A public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI. *Medical Image Analysis* 35 (2017), 250–269.

[111] Rupesh Mandal and Nupur Choudhury. 2016. Automatic video surveillance for theft detection in ATM machines: An enhanced approach. In *Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom'16)*. IEEE, Los Alamitos, CA, 2821–2826.

[112] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. 2018. Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 616–625.

[113] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. 2016. Convolutional oriented boundaries. In *Proceedings of the European Conference on Computer Vision*. 580–596.

[114] D. Martin, C. Fowlkes, D. Tal, and J. Malik. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th International Conference on Computer Vision*, Vol. 2. 416–423.

[115] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. 2011. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proceedings of the International Conference on Artificial Neural Networks*. 52–59.

[116] L. R. Medsker and L. C. Jain. 2001. *Recurrent Neural Networks: Design and Applications*. CRC Press, Boca Raton, FL.

[117] B. M. Mehtre, N. N. Murthy, S. Kapoor, and B. Chatterjee. 1987. Segmentation of fingerprint images using the directional image. *Pattern Recognition* 20, 4 (1987), 429–435.

[118] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, et al. 2015. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Transactions on Medical Imaging* 34, 10 (2015), 1993–2024.

[119] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ADE20K dataset. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*.

[120] Andrew Merlino, Daryl Morey, and Mark Maybury. 1997. Broadcast news navigation using story segmentation. In *Proceedings of the 5th ACM International Conference on Multimedia*. ACM, New York, NY, 381–391.

[121] Filippo Molinari, Guang Zeng, and Jasjit S. Suri. 2010. A state of the art review on intima-media thickness (IMT) measurement and wall segmentation techniques for carotid ultrasound. *Computer Methods and Programs in Biomedicine* 100, 3 (2010), 201–221.

[122] Takayasu Moriya, Holger R. Roth, Shota Nakamura, Hirohisa Oda, Kai Nagara, Masahiro Oda, and Kensaku Mori. 2018. Unsupervised segmentation of 3D medical images based on clustering and deep representation learning. In *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, Vol. 10578. International Society for Optics and Photonics, Bellingham, WA, 1057820.

[123] T. Nathan Mundhenk, Daniel Ho, and Barry Y. Chen. 2018. Improvements to context based self-supervised learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'18)*.

[124] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML'10)*. 807–814.

[125] Ahmed Nassar, Karim Amer, Reda El Hakim, and Mohamed El Helw. 2018. A deep CNN-based framework for enhanced aerial imagery registration with applications to UAV geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 1513–1523.

[126] Gerhard Neuhold, Tobias Ollmann, S. Rota Bulo, and Peter Kontschieder. 2017. The Mapillary Vistas dataset for semantic understanding of street scenes. In *Proceedings of the International Conference on Computer Vision (ICCV'17)*. 22–29.

[127] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1520–1528.

[128] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of the European Conference on Computer Vision*. 69–84.

[129] Christine M. Onyango and John A. Marchant. 2001. Physics-based colour image segmentation for scenes containing vegetation and soil. *Image and Vision Computing* 19, 8 (2001), 523–538.

[130] Anisha Pal, Shourya Jaiswal, Swarnendu Ghosh, Nibaran Das, and Mita Nasipuri. [n.d.]. SegFast: A faster SqueezeNet based semantic image segmentation technique using depth-wise separable convolutions. In *Proceedings of the 11th Indian Conference on Computer Vision, Graphics, and Image Processing (ICVGIP'18)*. ACM, New York, NY, 7.

[131] Nikhil R. Pal and Sankar K. Pal. 1993. A review on image segmentation techniques. *Pattern Recognition* 26, 9 (1993), 1277–1294.

[132] George Papandreou, Liang-Chieh Chen, Kevin P. Murphy, and Alan L. Yuille. 2015. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 1742–1750.

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

[134] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2536–2544.

[135] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. 2017. Large kernel matters—Improve semantic segmentation by global convolutional network. arXiv:1703.02719.

[136] Pedro O. Pinheiro, Ronan Collobert, and Piotr Dollár. 2015. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*. 1990–1998.

[137] Pedro O. Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. 2016. Learning to refine object segments. In *Proceedings of the European Conference on Computer Vision*. 75–91.

[138] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. 2017. The 2017 Davis Challenge on video object segmentation. arXiv:1704.00675.

[139] Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, et al. 2018. Diabetic retinopathy: Segmentation and grading challenge workshop. In *Proceedings of the IEEE International Symposium on Biomedical Imaging(ISBI'18)*.https://idrid.grand-challenge.org/organizers/

[140] Huafeng Qin and Mounim A. El-Yacoubi. 2017. Deep representation-based feature extraction and recovering for finger-vein verification. *IEEE Transactions on Information Forensics and Security* 12, 8 (2017), 1816–1829.

[141] P. Radau, Y. Lu, K. Connelly, G. Paul, A. Dick, and G. Wright. 2009. Evaluation framework for algorithms segmenting short axis cardiac MRI. *MIDAS Journal* 49 (2009).

[142] Anurag Ranjan, Varun Jampani, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. 2018. Adversarial collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. arXiv:1805.09806.

[143] Mahdyar Ravanbakhsh, Moin Nabi, Hossein Mousavi, Enver Sangineto, and Nicu Sebe. 2016. Plug-and-play CNN for crowd motion analysis: An application in abnormal event detection. arXiv:1610.00307.

[144] Mengye Ren and Richard S. Zemel. 2017. End-to-end instance segmentation with recurrent attention. arXiv:1605.09410.

[145] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. 91–99.

[146] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. 2016. Recurrent instance segmentation. In *Proceedings of the European Conference on Computer Vision*. 312–329.

[147] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*. 234–241.

[148] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. 2016. The Synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3234–3243.

[149] Brandon Rothrock, Ryan Kennedy, Chris Cunningham, Jeremie Papon, Matthew Heverly, and Masahiro Ono. 2016. Spoc: Deep learning-based terrain classification for Mars Rover missions. In *Proceedings of the AIAA SPACE 2016 Conference*. 5539.

[150] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323, 6088 (1986), 533.

[151] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*. 3856–3866.

[152] N. Senthilkumaran and R. Rajesh. 2009. Edge detection techniques for image segmentation—A survey of soft computing approaches. *International Journal of Recent Trends in Engineering* 1, 2 (2009), 250–254.

[153] Neeraj Sharma and Lalit M. Aggarwal. 2010. Automated medical image segmentation techniques. *Journal of Medical Physics/Association of Medical Physicists of India* 35, 1 (2010), 3.

[154] Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (2000), 888–905.

[155] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. 2016. Hierarchical image saliency detection on extended CSSD. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38, 4 (2016), 717–729.

[156] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. 2006. TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings of the European Conference on Computer Vision*. 1–15.

[157] Margarida Silveira, Jacinto C. Nascimento, Jorge S. Marques, André R. S. Marçal, Teresa Mendonça, Syogo Yamauchi, Junji Maeda, and Jorge Rozeira. 2009. Comparison of segmentation methods for melanoma diagnosis in dermoscopy images. *IEEE Journal of Selected Topics in Signal Processing* 3, 1 (2009), 35–45.

[158] Yan Song, Yuemei Zhu, Guangliang Li, Chen Feng, Bo He, and Tianhong Yan. 2017. Side scan sonar segmentation using deep convolutional neural network. In *Proceedings of the 2017 OCEANS–Anchorage Conference*. IEEE, Los Alamitos, CA, 1–4.

[159] Joes Staal, Michael D. Abràmoff, Meindert Niemeijer, Max A. Viergever, and Bram Van Ginneken. 2004. Ridge-based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging* 23, 4 (2004), 501–509.

[160] Tamás Szirányi, Károly László, László Czúni, and Francesco Ziliani. 1999. Object oriented motion-segmentation for video-compression in the CNN-UM. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology* 23, 2-3 (1999), 479–496.

[161] Khang Siang Tan and Nor Ashidi Mat Isa. 2011. Color image segmentation using histogram thresholding—Fuzzy c-means hybrid approach. *Pattern Recognition* 44, 1 (2011), 1–15.

[162] Orlando José Tobias and Rui Seara. 2002. Image segmentation by histogram thresholding using fuzzy sets. *IEEE Transactions on Image Processing* 11, 12 (2002), 1457–1465.

[163] Michael Treml, José Arjona-Medina, Thomas Unterthiner, Rupesh Durgesh, Felix Friedmann, Peter Schuberth, et al. 2016. Speeding up semantic segmentation for autonomous driving. In *Proceedings of the MLITS NIPS Workshop*.

[164] Wei-Chih Tu, Ming-Yu Liu, Varun Jampani, Deqing Sun, Shao-Yi Chien, Ming-Hsuan Yang, and Jan Kautz. 2018. Learning superpixels with segmentation-aware affinity loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 568–576.

[165] Jasper R. R. Uijlings, Koen E. A. Van De Sande, Theo Gevers, and Arnold W. M. Smeulders. 2013. Selective search for object recognition. *International Journal of Computer Vision* 104, 2 (2013), 154–171.

[166] Koen E. A. Van de Sande, Jasper R. R. Uijlings, Theo Gevers, and Arnold W. M. Smeulders. 2011. Segmentation as selective search for object recognition. In *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV'11)*. IEEE, Los Alamitos, CA, 1879–1886.

[167] Bram Van Ginneken, Mikkel B. Stegmann, and Marco Loog. 2006. Segmentation of anatomical structures in chest radiographs using supervised methods: A comparative study on a public database. *Medical Image Analysis* 10, 1 (2006), 19–40.

[168] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. V. Jawahar. 2018. IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments. arXiv:1811.10200.

[169] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv:1601.07140.

[170] David L. Vilarino, Diego Cabello, and Victor M. Brea. 2002. An analogic CNN-algorithm of pixel level snakes for tracking and surveillance tasks. In *Proceedings of the 2002 7th IEEE International Workshop on Cellular Neural Networks and Their Applications (CNNA'02)*. IEEE, Los Alamitos, CA, 84–91.

[171] Kai Wang, Boris Babenko, and Serge Belongie. 2011. End-to-end scene text recognition. In *Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV'11)*. IEEE, Los Alamitos, CA, 1457–1464.

[172] Guo-Qing Wei, Klaus Arbter, and Gerd Hirzinger. 1997. Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation. *IEEE Engineering in Medicine and Biology Magazine* 16, 1 (1997), 40–45.

[173] Xide Xia and Brian Kulis. 2017. W-Net: A deep model for fully unsupervised image segmentation. arXiv:1711.08506.
[174] Jieqiong Xu, Guoyu Wang, and Feifei Sun. 2013. A novel method for detecting and tracking vehicles in traffic-image sequence. In *Proceedings of the 5th International Conference on Digital Image Processing (ICDIP'13)*, Vol. 8878. 88782P.
[175] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. 2018. YouTube-VOS: A large-scale video object segmentation benchmark. arXiv:1809.03327.
[176] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. 2013. Saliency detection via graph-based manifold ranking. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*. IEEE, Los Alamitos, CA, 3166–3173.
[177] Fisher Yu and Vladlen Koltun. 2015. Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122.
[178] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. 2018. BDD100K: A diverse driving video database with scalable annotation tooling. arXiv:1805.04687.
[179] Jiangye Yuan, Shaun S. Gleason, and Anil M. Cheriyadat. 2013. Systematic benchmarking of aerial image segmentation. *IEEE Geoscience and Remote Sensing Letters* 10, 6 (2013), 1527–1531.
[180] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*. 818–833.
[181] Darko Zikic, Yani Ioannou, Matthew Brown, and Antonio Criminisi. 2014. Segmentation of brain tumor tissues with convolutional neural networks. In *Proceedings of the MICCAI Workshop on Multimodal Brain Tumor Segmentation Challenge (BRATS'14)*. 36–39.
[182] Xiaohang Zhan, Xingang Pan, Ziwei Liu, Dahua Lin, and Chen Change Loy. 2019. Self-supervised learning via conditional motion propagation. arXiv:1903.11412.
[183] Qi Zhang, Sally A. Goldman, Wei Yu, and Jason E. Fritts. 2002. Content-based image retrieval using multiple-instance learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML'02)*, Vol. 2. 682–689.
[184] Richard Zhang, Phillip Isola, and Alexei A. Efros. 2016. Colorful image colorization. In *Proceedings of the European Conference on Computer Vision*. 649–666.
[185] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. 2017. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing* 14, 2 (2017), 119–135.
[186] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*. 2881–2890.

Received July 2018; revised February 2019; accepted May 2019