
A SURVEY ON DEEP LEARNING METHODS FOR SEMANTIC IMAGE SEGMENTATION IN REAL-TIME

Georgios Takos

Mountain View, CA

georgios.takos@gmail.com

September 29, 2020

ABSTRACT

Semantic image segmentation is one of fastest growing areas in computer vision with a variety of applications. In many areas, such as robotics and autonomous vehicles, semantic image segmentation is crucial, since it provides the necessary context for actions to be taken based on a scene understanding at the pixel level. Moreover, the success of medical diagnosis and treatment relies on the extremely accurate understanding of the data under consideration and semantic image segmentation is one of the important tools in many cases. Recent developments in deep learning have provided a host of tools to tackle this problem efficiently and with increased accuracy. This work provides a comprehensive analysis of state-of-the-art deep learning architectures in image segmentation and, more importantly, an extensive list of techniques to achieve fast inference and computational efficiency. The origins of these techniques as well as their strengths and trade-offs are discussed with an in-depth analysis of their impact in the area. The best-performing architectures are summarized with a list of methods used to achieve these state-of-the-art results.

Keywords Semantic Image Segmentation · Real-Time Segmentation · Deep Learning · Convolutional Networks

1 Introduction

Semantic segmentation is one of the fastest growing areas in Computer Vision and Machine Learning. The availability of cameras and over devices have dramatically increased the interest in better understanding the context of the scene they are capturing and image segmentation is one of

the most important components of this process. When an image is analyzed the following levels of understanding are sought:

1. Classification, i.e., label the most prominent object of an image [1].
2. Classification with localization, i.e., extend the previous solution with a bounding box of the object in question.
3. Object detection, where multiple objects of different types are classified and localized [2].
4. Semantic segmentation, where every pixel in the image is classified and localized.
5. Instance segmentation, an extension to semantic segmentation where different objects of the same type are treated as distinct objects.
6. Panoptic segmentation, which combines semantic and instance segmentation such that all pixels are assigned a class label and all object instances are uniquely segmented.

The focus of this work is semantic image segmentation, where a pixel-level classification is targeted, and where image pixels which belong to the same object class are clustered together. An example of this pixel-level classification can be seen in Figure 1a and Figure 1b (see [3]). The original image on the left (Figure 1a) can be compared to the semantic segmentation target is the one on the right (Figure 1b), where all objects of interest have been classified.



(a) Original image

(b) Semantic segmentation ground truth

Figure 1: PASCAL VOC training images

Semantic segmentation plays an important role in diverse applications such as:

- Medical image diagnosis [4], [5].
- Autonomous driving [6], [7].
- Satellite image processing [8].
- Environmental analysis [9].
- Agricultural development [10].

- Image search engines [11].

In this paper we provide a comprehensive summary of the most recent developments in the area of semantic segmentation with a focus on real-time systems. Efficient techniques for semantic segmentation, where memory requirements and inference time are the main consideration, have not been sufficiently summarized before to the best of the author’s knowledge and have been instrumental for the increasing popularity of semantic segmentation across different fields.

This paper is organized as follows: in Section 2 the evolution of traditional image segmentation methods is summarized, followed by, in Section 3, a comprehensive summary of deep learning approaches. Section 4 summarizes the most seminal works on real-time systems, while analyzing the most effective techniques in terms of computational cost and memory load. The following Section enumerates the different datasets that have been used to benchmark different architectures, followed by a Section on the metrics used in the evaluation. The paper concludes with a summary of the performance of different real-time architectures.

2 History of Semantic Segmentation

One of the earlier approaches to semantic segmentation is thresholding [12], [13]. It attempts to divide the image to two regions, the target and the background. It works quite well in gray-level images that can be classified in a straightforward manner by using a single threshold. This technique has evolved by taking both local and global threshold values to better capture the image features.

A second technique involves clustering of pixels or regions with similar characteristics, where the image is split into K groups or clusters. All pixels are assigned a cluster based on a similarity metric that can involve the pixel features (e.g. color, gradient) as well as the relative distance [14]. Several popular segmentation techniques have been successfully applied, such as K-means [15], GMMs [16], mean-shift [17], and fuzzy k-means [18].

Edge detection methods [19], have used the fact the edges frequently represent boundaries that can help in segmenting the image. Different edge types have been used (e.g. step edges, ramp edges, line edges and roof edges). Most popular line edge detection methods include Roberts edge detection [20], Sobel edge detection [21], and Prewitt edge detection [22], which utilize different two-dimensional masks that when convolved with the image will highlight the edges.

A fourth approach looks at images like graphs, where each pixel is a vertex connected with all other pixels, with the weight of each edge measuring the similarity between the pixels. Similarity measures can use features such as distance, intensity, color, and texture to calculate the edge weights. Image segmentation is then treated like a graph partitioning problem, where graph segments are

partitioned based on the similarity of the groups [23] – [25]. An affinity matrix is computed and the solution to the graph cut problem is given by the generalized eigenvalue of the matrix.

Conditional Random Fields (CRF), a probabilistic framework that can be used to label and segment data, have been used extensively in image segmentation. In this framework, every pixel (that can belong to any of the target class) is assigned a unary cost, i.e., the price to assign a pixel to a class. In addition, a pair-wise cost is added that can model interactions between pixels. For example, zero cost can be assigned when two neighboring pixels belong to the same class, but non-zero cost when the pixels belong to different classes. The unary costs capture the cost for disregarding a class annotation, while the latter penalize non-smooth regions. The goal of the CRF is to find a configuration where the overall cost is minimized. An excellent explanation of CRFs is in [26], whereas applications in semantic segmentation can be found in [27], [28].

3 Deep Learning Approaches to Semantic Image Segmentation

3.1 Fully Convolutional Networks

Convolution networks were initially used for classification tasks (AlexNet [1], VGG [29], GoogLeNet [30]). These networks first processed the input image with several convolutional layers with increasing number of filters and decreasing resolution, with the last convolutional layer vectorized. The vectorized features were followed by fully connected layers that learn the probability distribution of the classes with a softmax output layer. In FCN [31], the fully connected layers that result in loss of the spatial information were removed from popular architectures ([1], [29], and [30]) and replaced with a layer that allow the classification of the image on a per-pixel basis (see Figure 2). The replacement of fully-connected layers with convolutional ones had two distinct advantages: (a) it allowed the same network architecture to be applied to an image of any resolution and, (b) convolutional layers have fewer parameters which allowed for faster training and inference. This novel approach resulted in state-of-the-art results in several image segmentation milestones and is assumed one of the most influential in the area.

3.2 Encoder-Decoder Architecture

In DeconvNet [32], the authors noted that the approach in [31] was leading to loss of information due to the the absence of real deconvolution and the small size of the feature map. They then proposed the architecture in Figure 3, where a multi-layer deconvolution network is learned. The trained network is applied to individual object proposals using fully-connected CRF to obtain instance-wise segmentations, which are combined for the final semantic segmentation. The encoder architecture is based on [29].

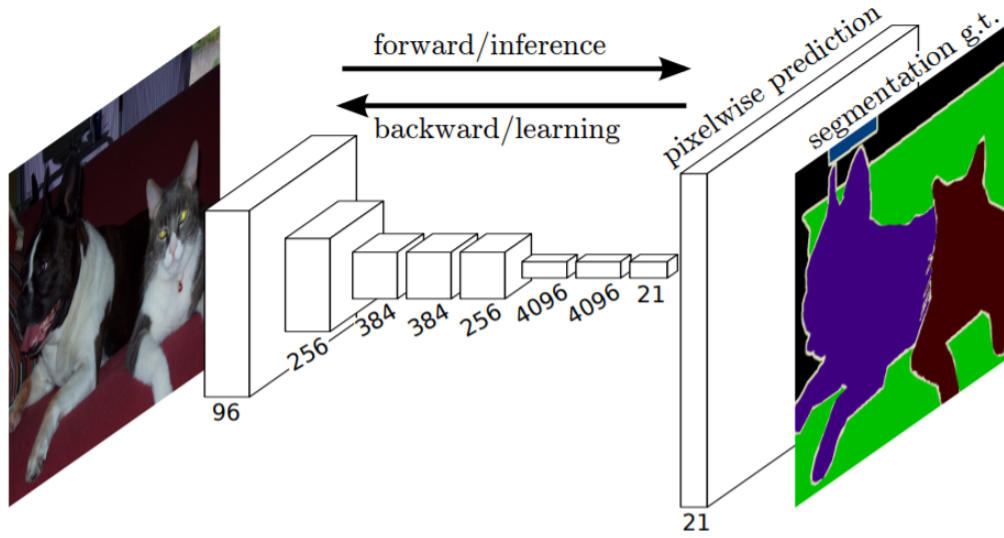


Figure 2: Fully Convolutional Network Architecture (from [31]).

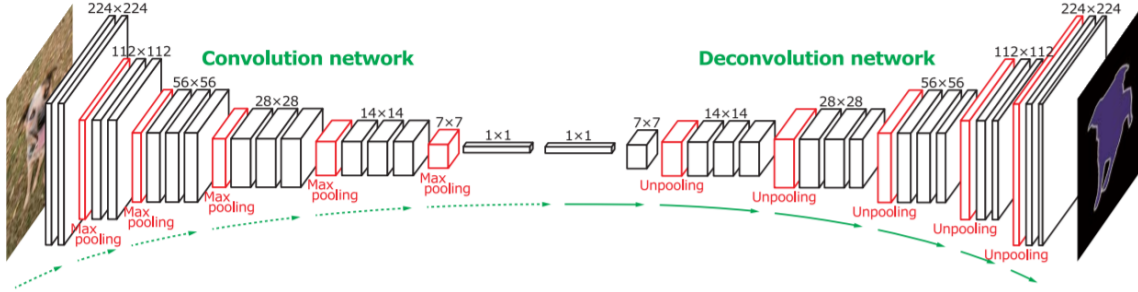


Figure 3: DeconvNet Architecture (from [32]).

In parallel to [32], decoder/encoder architectures were also used for medical applications [33]. The authors proposed an architecture that works well when little training data is available (30 images), which, with appropriate data augmentation can lead to state-of-the-art performance. In Figure 4 the decoder part on the left (contracting path according to the authors) downsamples the image while increasing the number of features. On the upsampling path, the opposite procedure is followed (i.e., increasing the image resolution while decreasing the number of features), while concatenating the corresponding encoder layer. They also proposed a weighted loss around different regions in order to achieve more accurate class separation.

A similar architecture to [33] was proposed in SegNet [34], where the authors used VGG [29] as the backbone encoder, removed the fully connected layers, and added a symmetric decoder structure. The main difference is that every decoder layer uses the max pooling indices from the corresponding encoder layer, as opposed to concatenating it. Reusing max-pooling indices in the decoding process has several practical advantages: (i) it improves boundary delineation , (ii) it

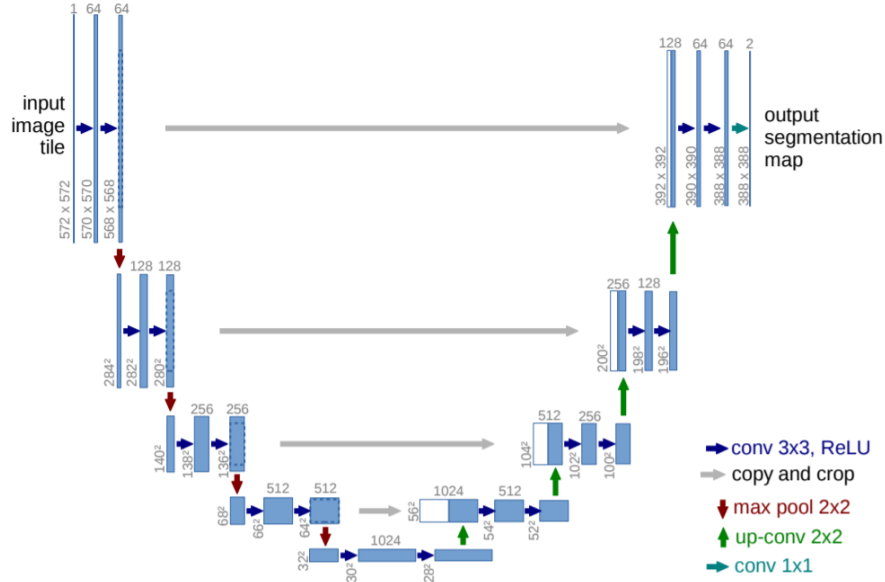


Figure 4: UNet Architecture (from [33]).

reduces the number of parameters enabling end-to-end training, and (iii) this form of upsampling can be incorporated into any encoder-decoder architecture. Although it was originally published in 2015, it originally received little traction until 2017, and has since become one of the most referenced works in semantic segmentation.

3.3 Conditional Random Fields with Neural Networks

Conditional Random Fields (CRFs), were one of the most popular methods in semantic segmentation before the arrival of deep learning. CRFs, however, due to their slow training and inference speeds, as well as the difficulty to learn their internal parameters, lost part of their appeal. On the other hand, CNNs by design are not expected to perform well in boundary regions, where two or more classes intersect, or can lose high-level information through the multiple processing stages.

The authors of [35] pooled the two approaches by combining the responses at the final neural network layer with a fully connected Conditional Random Field. This way, the model's ability to capture fine details is enhanced by incorporating the local interactions between neighboring pixels and edges. This work evolved into DeepLab [36], where several improvements were added (e.g., atrous spatial pyramid pooling), and several variants were proposed. The basic idea can be best explained by Figure 5: a fully convolutional network is used to get a coarse score map for the different classes. The image is then upsampled to its full resolution and the CRF is then deployed to better capture the object boundaries. DeepLab achieved state-of-the-art performance in multiple segmentation datasets with an inference time of 125ms or 8 frames per second (FPS).

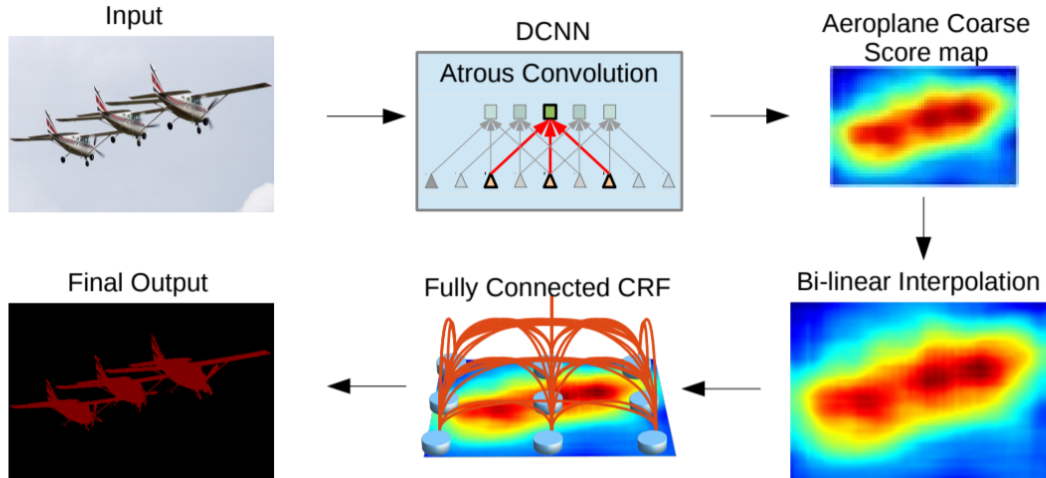


Figure 5: DeepLab Architecture (from [36]).

In the previous work, CRFs are not trained jointly with the fully convolutional network. This can lead to suboptimal end-to-end performance. In [37], the authors proposed to formulate CRF as an RNN to obtain a deep network that has desirable properties of both CNNs and CRFs. The two networks are then fully integrated and trained jointly to achieve top results on the PASCAL VOC 2012 segmentation benchmark [3].

3.4 Feature Fusion

Semantic segmentation involves the task of classifying an image on a pixel level. A lot of the techniques in the area have focused on getting the details of the image right, whereas the context at different stages gets lost. The authors in [38] suggest enhancing the performance of fully convolutional networks by adding global context to help clarify local confusions. In particular, they propose using the average feature for each layer to augment the features at each location, and thus use the combined feature map to perform segmentation. The effect of global context can be significant; a lot of the misclassified pixels using traditional fully convolutional networks can be recovered when the global context clarifies local confusion and, as a result, a smoother segmentation output is produced.

The authors in [39] proposed the Enhanced Semantic Segmentation Network (ESSN), that upsamples and concatenates the residual feature maps from each convolutional layer in order to maintain features from all stages of the network (as seen in Figure 6). In [40], there is a downsampling stage that extracts feature information, followed by an upsampling part to recover the spatial resolution. The features of the corresponding pooling and unpooling layers are upsampled and concatenated, before the final prediction stage that produces the segmentation output. This fusion at multiple levels

of the features maps was evaluated on three major semantic segmentation datasets and achieved promising results.

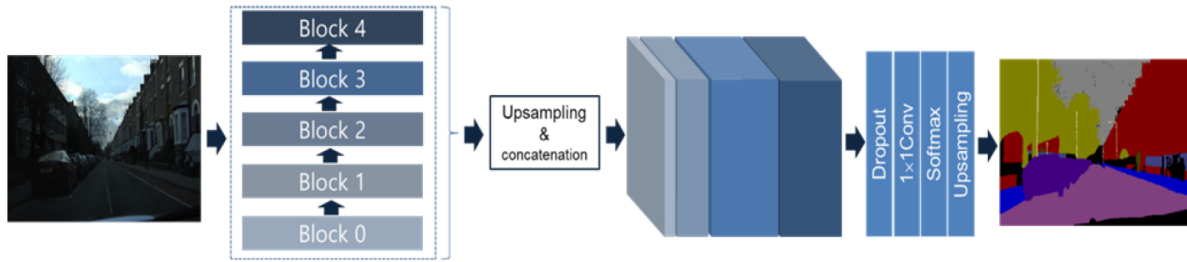


Figure 6: Enhanced Semantic Segmentation Network Architecture (from [39]).

3.5 Generative Adversarial Networks

Generative Adversarial Networks (GANs) were originally introduced in [41] as a generative model for unsupervised learning, where the model learns to generate new data with the same statistics as the training set. Its first demonstration was on images, where the artificially generated images looked very similar to those of the training set. Since then, GANs have had quite an impact in diverse areas such as astronomical images [42], 3D object reconstruction [43], and image super-resolution [44].

The idea to apply GANs in semantic segmentation was first introduced in [45], where the authors used two different networks. First, a segmentation network that took the image as an input and generated per-pixel predictions much like the traditional CNN approaches described earlier in this work, and, second, an adversarial network that discriminates segmentation maps coming either from the ground truth or from the segmentation network. The adversarial network takes as input the image, the segmentation ground truth, and the segmentation network output and outputs a class label (1 for ground truth and 0 for synthetic). An adversarial term is added to the cross-entropy loss function. The adversarial term encourages the segmentation model to produce label maps that cannot be distinguished from ground-truth ones and lead to improved labeling accuracy in the Stanford Background and PASCAL VOC 2012 datasets.

In [46], the authors proposed a semi-supervised framework – based on Generative Adversarial Networks (GANs) – which consists of a generator network to provide extra training examples to a multi-class classifier, acting as discriminator in the GAN framework, that assigns every sample a label from the K possible classes or marks it as a fake sample (extra class) as seen in Figure 7. The underlying idea is that adding large fake visual data forces real samples to be close in the feature space, which, in turn, improves multiclass pixel classification.

The authors of [47], applied semantic segmentation with GANs in medical images. In a similar fashion to [45], the adversarial network takes as inputs the original image, the segmentation

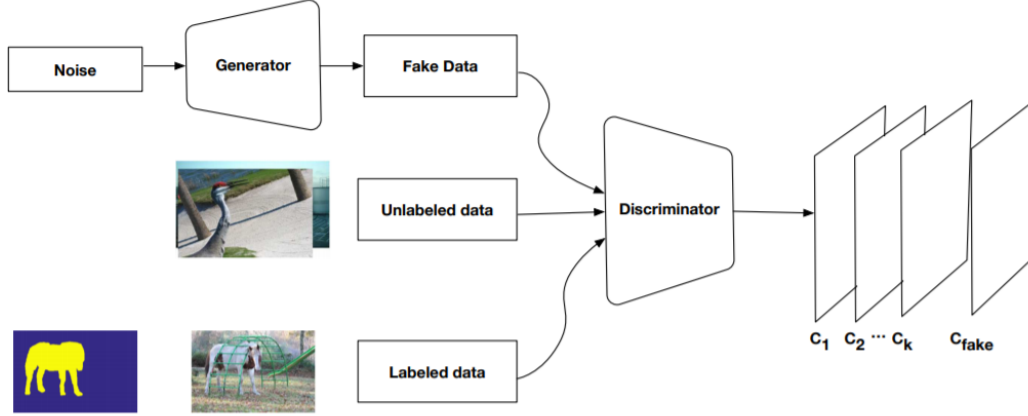


Figure 7: Semi-Supervised Convolutional GAN Architecture (from [46]).

network output, and the ground truth, and optimize a multi-scale loss function that uses the mean absolute error distance in a min-max fashion. The segmentation network consists of four layers of convolutional stages as in [33], tailored to work with the limited training data sets, and the network significantly outperforms [33].

3.6 Recurrent Neural Nets (RNNs)

RNNs [48] have been widely used for sequential tasks. In [49] the authors proposed ReSeg, which was based on the recently introduced ReNet model for image classification [50]. The latter was tailored to semantic segmentation tasks by transforming each ReNet layer. In particular, each ReNet layer is composed of four RNN (GRUs, see [48]) that sweep the image horizontally and vertically in both directions, encoding patches or activations, and providing relevant global information. Moreover, ReNet layers are stacked on top of pre-trained convolutional layers, benefiting from generic local features. Upsampling layers follow ReNet layers to recover the original image resolution in the final predictions. The network architecture can be better understood by looking at Figure 8. The first 2 RNNs (blue and green) are applied on small patches of the image, their feature maps are concatenated and fed as input to the next two RNNs (red and yellow) which emit the output of the first ReNet layer. Two similar ReNet layers are stacked, followed by an upsampling layer and a softmax nonlinearity.

Another interesting application to image segmentation was in [51]. There the authors looked at the problem of video segmentation, where consecutive video frames are segmented. One approach would be to independently segment each frame, but this looks like an inefficient approach due to the highly correlated nature of video frames. The authors suggested to incorporate the temporal information by adding an LSTM [48], a type of RNN that can efficiently handle long

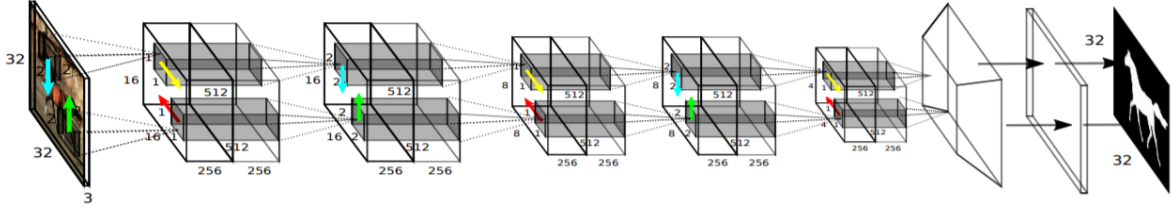


Figure 8: ReSeg Network Architecture (from [49]).

time dependencies, at different stages in the network and they reported significant performance improvement over their CNN counterparts.

3.7 Panoptic Segmentation

Panoptic segmentation [59], the task that tries to combine semantic and instance segmentation such that all pixels are assigned a class label and all object instances are uniquely segmented, has shown very promising results [60], [61]. The task to provide a coherent scene segmentation incorporating both semantic and instance segmentation seems to be leading to state-of-art results in semantic segmentation over several benchmark data sets as shall be seen later in this report.

3.8 Attention-based Models

Attention in deep learning was first introduced in the field of machine translation [52]. The attention mechanism captured long-range dependencies in an effective manner by allowing the model to automatically search for parts of the source sentence that are relevant to predicting a target word.

One interesting way that attention was introduced in semantic segmentation was to incorporate multi-scale features in fully convolutional networks. Instead of the traditional method of feeding multiple resized images to a shared deep network, the authors of [53] proposed an attention mechanism that learns to softly weight the multi-scale features at each pixel location. The convolutional neural network is jointly trained with the attention model as seen in Figure 9. As a result, the model learns to scale different size images in an appropriate fashion so that more accurate segmentation is achieved.

In a similar fashion, [54] tried to address the spatial resolution loss of fully convolutional networks by introducing the feature pyramid attention module. The latter combines the context features from different scales in order to improve classification performance of smaller objects. Attention-aided semantic segmentation networks have been widely used in a variety of applications [55]–[58].

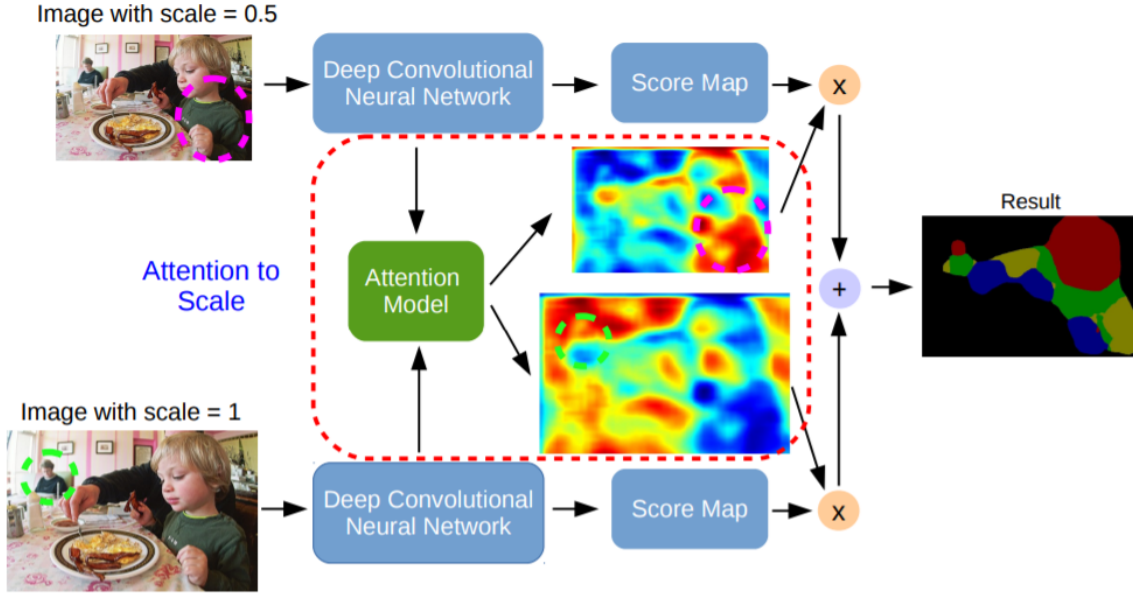


Figure 9: Scale-aware Semantic Image Segmentation Architecture (from [53]).

4 Real-Time Deep Learning Architectures for Semantic Image Segmentation

Deep learning based semantic segmentation accuracy has improved significantly from the early approaches. For example, [31] achieved 65% mean intersection over union (mIoU) in the Cityscapes data set [62] and 67% mIoU in the PASCAL VOC 2012 data set [63]. More recent architectures have outperformed these initial results quite significantly. The authors of HRNet [64], have build a hierarchical scheme to use images of different scales with an appropriate attention mechanism, like we saw at the end of the previous section. This approach achieves >85% mIoU in the Cityscapes data set. On the other hand, the authors in [65], have used a combination of data augmentation and self-training [66] – that uses noisy labels generated from a model trained on a much smaller labeled data set – to get >90% mIoU in the PASCAL VOC 2012 data set.

Computation efficiency, however, is also of paramount importance in several areas like self-driving cars and segmentation on mobile devices, where inference requirements are quite limiting. Computational/memory cost and inference time have to be taken into account when designing a real-time system. In this section, we will go over an exhaustive list of the techniques to build such a system and explain how these improvements were implemented in the literature.

4.1 Fast Fourier Transform (FFT)

The well-known convolution theorem [67], states that under suitable conditions the Fourier transform of a convolution of two signals is the pointwise product of their Fourier transforms. The authors in

[68] exploited that fact to improve the training and inference time of convolutional networks. A convolution of an image of size $n \times n$ with a kernel of size $k \times k$ will take $\mathcal{O}(n^2 * k^2)$ operations using the direct convolution, but the complexity can be reduced to $\mathcal{O}(n^2 \log n)$ by using the FFT-based method. Some additional memory is required to store the feature maps in the Fourier domain, which is insignificant compared to the overall memory requirements of a deep neural network.

In [69], the training and inference algorithms were developed based on FFT and achieved reduced asymptotic complexity of both computation and storage. The authors claim a 1000x reduction in the required number of ASIC cores, as well as a 10x faster inference with a small reduction in accuracy.

4.2 Pruning

Storage and memory requirements of a neural network can also be reduced by pruning the redundant weights. In [70], suggested a three-step approach: first train the network to learn which connections are important, then prune the unimportant connections, and, finally, retrain the network to fine tune the weights of the remaining connections. The number of connections was, therefore, reduced by 9x to 13x with little performance degradation.

The work in [71], focuses on channel pruning for semantic segmentation networks. They reduce the number of operations by 50% while only losing 1% in mIoU using the following strategy: pruning convolutional filters based on both classification and segmentation tasks. This is particularly useful in cases where the network backbone was transferred from an architecture originally built for classification tasks as we have seen earlier in this report. Scaling factors for each convolutional filter are computed based on both tasks and the pruned network is used for inference.

Network pruning is a very active area to improve performance in convolutional neural nets and semantic segmentation, see [72] and [73], where channel pruning methods can lead to significant compression and speed-up on various architectures that would work on multiple tasks (classification, detection, and segmentation), by either reducing the number of channels on a layer by layer fashion solving a LASSO regression optimization problem, or by pruning the backbone network before transferring it to the segmentation network.

4.3 Quantization

Another way to make the network more efficient is by reducing the number of bits required to represent each weight. Typically 32 bits are reserved for the representation of weights. 32-bit operations are slow and have large memory requirements. In [74] the authors suggest, among others, to reduce the weight representation to 5 bits, while limiting the number of effective weights by having multiple connections share the same weight, and then fine-tune those shared weights, thus reducing storage requirements.

In Bi-Real Net [75], the authors investigate the enhancement of 1-bit convolutional neural networks, where both the weights and the activations are binary. The performance of these 1-bit CNNs is improved by taking the real-valued output of the batch-normalization layer before the binary activation and connecting it to the real-valued activation of the next block. Thus the representational capability of the proposed model is much higher than that of the original 1-bit CNNs, with only a negligible computational cost.

4.4 Depthwise Separable Convolutions

The two previous methods aim at reducing the network size, by either pruning unnecessary components or compressing the weight information. Sifre in his Ph.D. thesis [76] introduced a novel method to make 2-dimensional convolutions a lot more computationally efficient called depthwise separable convolution. This idea was picked up by Xception [77] and MobileNets [78] that used slightly modified versions of the original idea to greatly improve the efficiency of their relative architectures. In a regular convolutional layer, the computation complexity depends on (a) the input/output feature map of size $D \times D$ (square feature map is assumed for simplicity), (b) the number of inputs channels M , (c) the number of output channels N , and (d) the spatial dimension of the kernel K . The overall computation requires $D^2 \times K^2 \times M \times N$ multiplications.

In depthwise separable convolutions, the convolution with the filter of size $K \times K \times M \times N$ is broken into two parts. First, a depthwise convolution of a single filter per channel, i.e., of size $K \times K$ for all M input channels, and (b) a pointwise convolution that uses 1×1 convolutional filters to generate the appropriate output channel dimension. The first operation requires $D^2 \times K^2 \times M$, while the second $D^2 \times M \times N$. The computational improvement is of the order $\max(\mathcal{O}(N), \mathcal{O}(D^2))$, which can be quite significant especially when the filter size or depth increases.

4.5 Dilated Convolutions

In their seminal work [79], the authors introduced dilated convolution to expand the effective receptive field of kernels by inserting zeros between each pixel in the convolutional kernel. As seen on the left side of Figure 10, a 3×3 kernel will cover nine pixels. However, if a dilation rate of 2 is introduced, then the eight outer pixels will expand to cover twenty-five pixels, by skipping a pixel (see middle of Figure 10). If the dilation rate further doubles, the coverage will then be of eighty-one pixels as seen on the right of Figure 10. In summary, a kernel of size $K \times K$ with a dilation rate of N will cover $(N - 1) * K \times (N - 1) * K$ pixels for an expansion of $(N - 1) \times (N - 1)$. In semantic segmentation tasks, where context is critical for the network accuracy, dilated convolutions can expand the receptive field exponentially without increasing the computation cost. By stacking multiple convolutional layers with different dilation rates, [79] managed to capture image context

of increasing receptive fields and was able to significantly improve segmentation performance of previous state-of-the-art works.

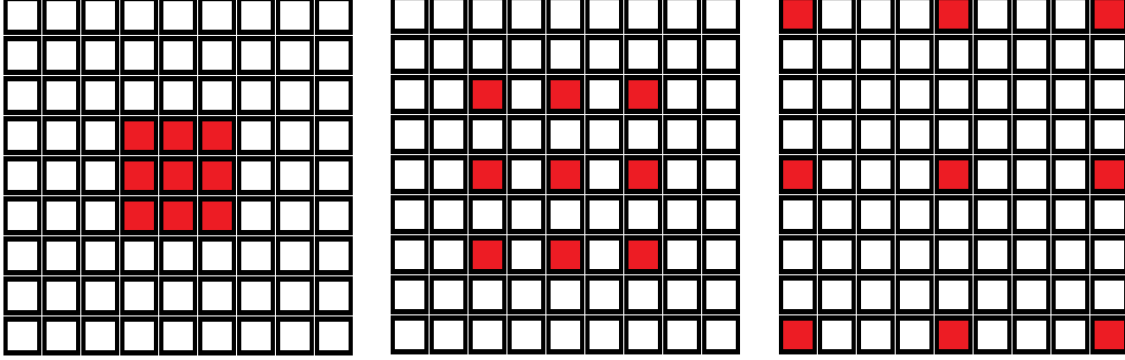


Figure 10: Schematic of 3×3 dilated convolutional kernel. Left: dilation rate = 1, Center: dilation rate = 2, Right: dilation rate = 4.

In [80] a new convolutional module was introduced, the efficient spatial pyramid (ESP). ESPNet combines dilated convolution with the depthwise separable convolutions of the previous subsection. In other words, the authors formed a factorized set of convolutions that decompose a standard convolution into a point-wise convolution and a spatial pyramid of dilated convolutions. ESPNet had among the smallest number of parameters of similar works while maintaining the largest effective receptive field. This work is especially interesting since it introduced several new system-level metrics that help to analyze the performance of CNNs.

4.6 Width and Resolution Multipliers

In [78], the authors explored several ways to further reduce the network complexity. They introduced two hyperparameters: (a) the width multiplier that would produce thinner models and (b) the resolution multiplier that would reduce input resolution. In the former case, the authors scaled down the computational requirements of every layer in a uniform fashion by scaling the number of input and out channels by a factor α . From the analysis of the depthwise separable convolutions, it can be seen that the original computational complexity of $D^2 \times K^2 \times M + D^2 \times M \times N$ becomes $D^2 \times K^2 \times M \times \alpha + D^2 \times M \times N \times \alpha^2$, for an overall reduction of somewhere between α and α^2 . The basic idea is to find an appropriate scaling factor to define a new smaller model with a reasonable accuracy, latency, and size trade off.

On the other hand, the resolution multiplier ρ can scale the input image dimensions by a factor of ρ^2 leading to an overall computational cost of $D^2 \times K^2 \times M \times \rho^2 + D^2 \times M \times N \times \rho^2$. Again

this reduction process should be optimized with the accuracy, latency, and size in mind. The two methods can be combined for further improvements.

4.7 Early Downsampling

A similar idea was presented in [81], where a number of design choices were discussed based on the authors experimental results and intuitions. In particular, very large input frames are very expensive computationally and it is a good idea to downsample these frames in the early stages of the network, while keeping the number of features relatively low. This downsampling does not have a severe impact in the overall performance because the visual information is typically highly redundant and can be compressed into a more efficient representation. Furthermore, it is noted that the first few layers do not really contribute to the classification task but rather provide useful representations for the subsequent layers. On the other hand, filters operating on downsampled images have a larger receptive field and can provide more context to the segmentation task.

Since the downsampling can lead to loss of spatial information like exact edge shape, ENet follows the paradigm set in SegNet [34], where the indices of elements chosen in max pooling layers are stored and used to produce sparse upsampled maps in the decoder, therefore, partially recovering spatial information, with small memory requirements.

4.8 Smaller Decoder Size

Another design choice discussed in [81] was the fact that, in the typical encoder/decoder structure of a semantic segmentation network, the two subcomponents do not have to be symmetric. Encoders need to be deep in order to capture the features in a similar fashion to their classification counterparts. Decoders, however, have one main task: to upsample the compressed feature space in order to provide pixel-level classification. The latter can be achieved with a much less deep architecture, providing significant computational savings.

4.9 Efficient Grid Size Reduction

The authors in [82] noticed that because the pooling operation can lead to a representational bottleneck, it is typically compensated by increasing the number of channels used before the pooling operation. Unfortunately, this means that the doubling of the filters is effectively dominating the computational cost. Reversing the order of the convolution/pooling operations would definitely improve the computation speed, but would not help with the representational bottleneck. What the authors suggest is to perform pooling operation in parallel with a convolution of stride 2, and concatenate the resulting filter banks. This technique allowed the authors of [81] to speed up inference time of the initial block by a factor of 10.

4.10 Drop Bias Terms

Bias terms do not have significant impact on the overall performance of a semantic segmentation network and are typically dropped.

4.11 Stack Multiple Layers with Small Kernels

Total computational cost increases with the square of the kernel size. In [29], it was argued that having multiple convolutional layers with small kernel size is superior to having a single layer with a larger kernel for two reasons: (a) by stacking three 3×3 convolutional layers correspond to the same effective receptive field of a 7×7 layer while reducing the number of parameters to almost half and, (b) by incorporating three non-linear rectification layers instead of a single one, the decision function is made more discriminative.

4.12 Channel Shuffle Operation

Grouped convolutions were first introduced in [1] to distribute the model over multiple GPUs. It uses multiple convolutions in parallel in order to derive multiple channel outputs per layer. [83] showed that the use of grouped convolutions can improve accuracy in classification tasks. However, this architecture become less efficient when applied to much smaller networks, where the performance bottleneck is the large number of dense 1×1 convolutions. The authors in [84] propose a novel channel shuffle operation to overcome this difficulty. In particular, on the left side of Figure 11 a typical group convolution can be seen with two stacked layers of convolutions and an equal number of groups. If a group convolution is allowed to obtain data from different groups then the input and output channels are fully related (see middle portion of Figure 11). However, the operation above can be efficiently implemented by following the process at the right of Figure 11. By adding a channel shuffle operation the output channel dimension is reshaped, transposed and the flattened before being fed to the subsequent layer. Channel shuffle reduces the number of operations by a factor of g , which is the number of groups.

4.13 Two Branch Networks

Downsampling the original image can lead to significant improvements in the inference time of a semantic segmentation architecture, but it can lead to significant loss of spatial detail. Two branch networks try to reconcile this by introducing two separate branches: (a) a relatively shallow branch that uses the full resolution image that captures the spatial details, and (b) a deeper branch with a downsampled image that will be able to learn efficiently the features for an effective classification outcome. The two branches can share layers as to further improve computational complexity (see [85]) or can have different backbone architectures before being aggregated [86]. The latter work

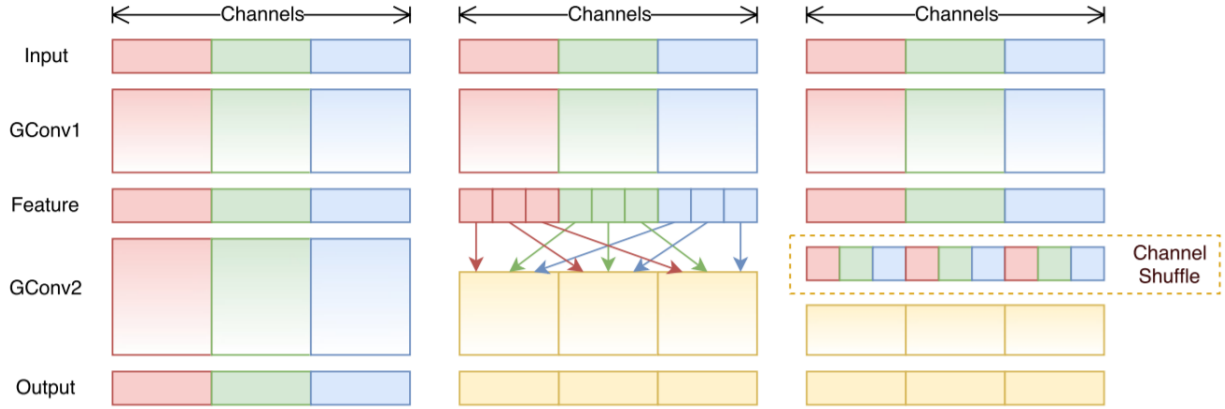


Figure 11: Channel Shuffle Architecture (from [84]).

is composed of a detail branch that uses wide channel and shallow layers to capture the low-level details, a semantic branch with narrow channels and deep layers to maintain the high-level context, and an aggregation layer to fuse the two kinds of features. As a result, BiSeNet-V2 achieves, arguably, the highest inference speed in semantic segmentation tasks (156 frames per second) while maintaining one of the best mIoU performance.

4.14 Other Design Choices

Apart from the computationally efficient methods presented in this section so far, there are a handful of other good design choices that would help maintain good performance despite using a lightweight architecture. For example, a common theme in many of the papers is batch normalization [87] that allows for faster and more accurate training process. Also the choice of the activation function can be significant. ReLU is the nonlinearity that most works use in the area, but several researchers have reported improved results with parametric ReLU (PReLU). Finally, regularization [88] can help avoid overfitting since in many applications the input image dimension is small compared to the number of parameters in a segmentation deep neural network.

5 Semantic Segmentation Data Sets

Several data sets have been generated in order to facilitate faster growth in key areas of semantic segmentation as well as establish performance benchmarks. Table 1 summarizes several of the image sets that have been annotated on a pixel level. It contains diverse datasets that were originally developed for classification tasks, as well as more specialized image sets that are appropriate for specific applications (e.g., self-driving and motion-based segmentation) covering a wide range of scenes and object categories with pixel-wise annotations. Additional information for each of these datasets will be provided in the remainder of this section.

Data Set	Images	Classes	Year
COCO	164K	172	2017
ADE20K	25.2K	2693	2017
Cityscapes	25K	30	2016
SYNTHIA	13K	13	2016
PASCAL Context	10.1K	540	2014
SIFT Flow	2.7K	33	2009
CamVid	701	32	2008
KITTI	203	13	2012

Table 1: Semantic Segmentation Data Set Summary

5.1 Common Objects in Context (COCO)

Common Objects in Context (COCO) [89] is a large-scale object detection, segmentation, and captioning dataset. It is one of the most extensive datasets available with 330K images of which half are labeled. Semantic classes can be either things (objects with a well-defined shape, e.g. car, person) or stuff (amorphous background regions, e.g. grass, sky). There are 80 object categories, 91 stuff classes, 1.5 million object instances, and due to the size of the data set it is considered one of the most challenging ones for image segmentation tasks. As a result, the COCO leader board [90] for semantic segmentation consists of only five entries with some of the most seminal works in the area occupying the top spots.

COCO-Stuff [91] augments all images of the COCO 2017 data set with pixel-wise annotations for the 91 stuff classes. The original COCO data set already provided outline-level annotation for the 80 thing classes, but COCO-stuff completed the annotation for more complex tasks such as semantic segmentation

5.2 PASCAL Visual Object Classes (VOC)

One of the most popular image sets is the PASCAL Visual Object Classes (VOC) [3] that can be used for classification, detection, segmentation, action classification, and person layout. The data are available in [63], have been annotated, and are periodically updated. For the image segmentation challenge the data include 20 classes categorized in every-day objects (airplane, bicycle, bird, boat, etc.). The training set consists of 1464 images and the validation set consists of 1449 images. The test set is reserved for evaluation in the PASCAL VOC Challenge, a competition that started in 2005 and it had its most recent data set in 2012. It is a generic data set that includes a variety of scenes/objects and, as a result, it is regularly used to evaluate novel image segmentation approaches. An example of an image and its semantic segmentation can be seen in Figure 1a and Figure 1b respectively.

Several extensions have been made to the former image set, most notably PASCAL Context [92] and PASCAL Part [93]. The former annotates the same image with over 500 classes, while the latter breaks down the original objects into several parts and annotates them. Two other PASCAL extensions are: (a) the Semantic Boundaries Data set (SBD) [94], and (b) the PASCAL Semantic Parts (PASParts) [95].

5.3 ADE20K

ADE20K [96] was developed by the MIT computer vision lab. The authors saw that the datasets available at the time were quite restrictive in the number and type of objects as well as the kinds of scenes. As a result, they collected a data set of 25K images that has densely annotated images (every pixel has a semantic label) with a large and an unrestricted open vocabulary of almost 2700 classes. The images in this data set were manually segmented in great detail, covering a diverse set of scenes, object and object part categories. A single expert annotator, providing extremely detailed and exhaustive image annotations without suffering from annotation inconsistencies common when multiple annotators are used. The detail in the annotation can be seen in Figures 12a and 12b. On average there are 19.5 instances and 10.5 object classes per image.

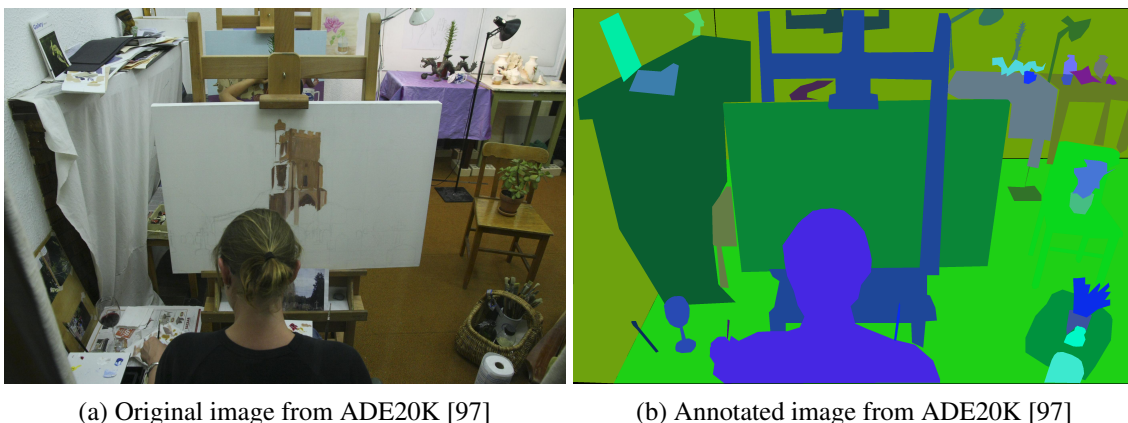


Figure 12: ADE20K training images

For their scene parsing benchmark [97], they selected the top 150 categories ranked by their total pixel ratios and they use the following metrics: (a) pixel accuracy, (b) mean accuracy, (c) mean IoU, and (d) weighted IoU. A little over 20K images were used for the training set, 2K images for the validation, and the remainder was reserved for testing. Stereo video sequences recorded in streets from 50 different cities and annotations involved 30 diverse classes.

5.4 Cityscapes

The Cityscapes data set [98] focuses on visual understanding of complex urban street scenes. It has 25K images, 5K of which have high quality pixel-level annotations, while 20K additional images have coarse annotations (i.e., weakly-labeled data) as be seen in Figures 13a and 13b, respectively.



(a) Fine annotation example from Cityscapes [99] (b) Coarse annotation example from Cityscapes [99]

Figure 13: Cityscapes training images

Their benchmark suite (found in [99]) involves, among others, a pixel-level semantic labeling task with over 200 entries. It is considered the most diverse and challenging urban scene data sets and, as a result, it is very popular performance evaluation tool.

5.5 SYNTHIA

The SYNTHIA data set [100] is another collection of urban scene images that focuses on self-driving applications. The authors generated realistic synthetic images with pixel-level annotations and tried to address the question of how useful such data can be for semantic segmentation. 13K urban images were created with automatically generated pixel level annotations from 13 categories (e.g. sky, building, road). It was concluded that, when SYNTHIA is used in the training stage together with publicly available real-world urban images, the semantic segmentation task performance significantly improves. An example of a synthetic image from SYNTHIA can be seen in Figure 14 as well as the general view of the city used for the image generation.

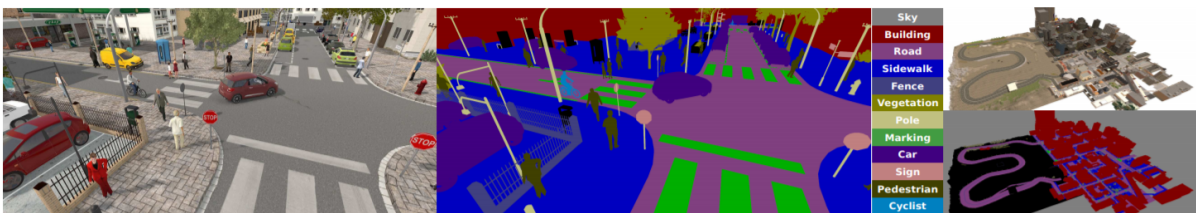


Figure 14: Sample synthetic image from SYNTHIA with its semantic label and a general view of the city (from [100]).

5.6 SIFT Flow

SIFT Flow [102] is data set that processed a subset of LabelMe images [103] in ordered to provide accurate pixel-level annotation of 2688 frames. The top 33 object categories (with the most labeled pixels) were selected mostly from outdoor scenes. The images were relatively small in size (256×256 pixels), and they were generated to evaluate the scene parsing algorithm of the authors.

5.7 CamVid

CamVid [104] is another urban scene dataset that includes four high-definition video sequences captured at 960×720 pixels at 30 frames per second. The total duration of the videos was a little over 22 minutes or around 40K frames. Of the latter, 701 were manually labeled with 32 object classes. Interestingly, the annotation effort took approximately 230 man-hours for an average annotation time of just under 20 minutes. Every annotated image was inspected and confirmed by a second person for accuracy.

5.8 KITTI

The last dataset we will be looking at is KITTI [105] that is quite popular in the self-driving research, since it contains not only camera images, but also laser scans, high-precision GPS measurements and IMU accelerations from a combined GPS/IMU system, sensor data that most autonomous vehicle efforts typically collect. The data were collected while driving in and around Karlsruhe, Germany and it contains over 200 fully annotated images from 13 different classes [106]. Their semantic segmentation benchmark contains 14 entries, where performance metrics include runtime and environment information to the time-sensitive target applications.

6 Metrics

In this section we will be summarizing the basic metrics used to evaluate different semantic segmentation approaches. They either look at the accuracy of the segmentation output (i.e., how close it is to the ground truth) or the efficiency of the approach (i.e., inference time and memory usage).

6.1 Confusion matrix

In a segmentation task where there is a total of C classes, the confusion matrix is a $C \times C$ table, where the element in position (i, j) represents the count of pixels that should belong to class i but where classified to belong to class j . A good model would result in a confusion matrix that has high counts in its diagonal elements (i.e., correctly classified pixels).

6.2 Normalized confusion matrix

It is derived from the confusion matrix, but every entry is normalized by dividing it to the total number of the predicted class j . This way all entries are in the range $[0, 1]$.

6.3 Accuracy

Accuracy, or global accuracy, is the ratio of the correctly classified pixels over the total pixels. It can be derived from the confusion matrix by dividing the sum of the diagonal elements to the total pixels in the image. Accuracy can be misleading especially when the classes under consideration are not balanced. For example, if 95% of the pixels are of one class (typically background), a trivial model always predicting this class will lead to 95% accuracy, which does definitely not capture the dependencies of the segmentation task.

6.4 Mean accuracy

It is defined as the ratio of correctly classified pixels in each class to total pixels averaged over all classes.

6.5 Mean intersection over union

Mean intersection over union (mIoU) is a metric that addresses the class imbalance weakness of the accuracy metric. In particular, it compares the pixel-wise classification output of a model with the ground truth and finds their intersection and union (i.e., how many pixels were correctly classified as class i for all classes i , as well as how many pixels were either classified or were annotated as class i for all classes i). The ratio of the intersection over the union (summed over all classes) is the mIoU or Jaccard index. It is robust to class imbalances and is, arguably, the most popular metric when evaluating semantic segmentation tasks.

6.6 Weighted intersection over union

This is a small variation of the previous metric to account for the number of pixels per class. It calculates the weighted average of the IoU for each class, weighted by the number of pixels in the class.

6.7 Precision

Precision for class i is defined as the proportion of pixels classified as i that were correctly classified. An average precision metric can be defined accordingly for multiple classes.

6.8 Recall

Recall for class i is defined as the proportion of the actual pixels of class i that were correctly classified. Similarly, an average recall metric can be defined accordingly for multiple classes.

6.9 F1-score

F1-score is aggregating the precision/recall metrics by calculating their harmonic mean. It combines features of both and provides information for both types of errors.

6.10 Frames per second

All previous metrics measure at the accuracy of the model output, but do not capture the efficiency of the method. One important metric to capture is the inference speed of a network, i.e., the execution time measured in frames per second (fps). It is the inverse of the time to run inference of a new image on a fully trained network. In most real time applications, an fps of 30 or more is required, usually to outperform a typical video frame rate.

6.11 Memory usage

Memory usage is a measure of the network size. It can either be measured in number of parameters (for a deep neural network approach), or the memory size to represent the network, or the number of floating point operations (FLOPs) required to run the model.

7 Performance Summary

In this section we will provide summary tables of the best performing models in semantic segmentation. Most papers get evaluated on a subset of the data sets provided earlier in this report and, for most works, computational efficiency is not an critical aspect of the design. As a result, it was decided to summarize the best performing models on the Cityscapes data set [98], which has been popular with most real-time architectures as an evaluation benchmark. Table 2 summarizes the top ten performing models with respect to the mIoU with a short summary of the methods used to achieve these results. Anonymous submissions were not included in this section despite occupying some of the top performing spots in the benchmark evaluation. As can be seen in Table 2, most entries were published over the past few months, suggesting a very competitive landscape with remarkably fast progress.

Table 3 ranks real-time semantic segmentation works where the performance metric is inference speed (i.e., frames per second (FPS)). Three of the top ten positions are occupied by a single paper [85], which clearly demonstrates the performance/efficiency trade-offs. However, as this table

Model	mIoU	Methods	Year
Hierarchical Multi-Scale Attention for Semantic Segmentation [64]	85.4	Hierarchical Attention	2020
Naive-Student (iterative semi-supervised learning with Panoptic-DeepLab) [107]	85.2	Pseudo-Label Prediction, Data Augmentation	2020
Object-Contextual Representations for Semantic Segmentation [108]	84.5	Coarse Soft Segmentation, Weighted Aggregation	2020
Panoptic-DeepLab [60]	84.5	Panoptic Segmentation	2020
EfficientPS: Efficient Panoptic Segmentation [61]	84.2	Panoptic Segmentation	2020
Axial-DeepLab [109]	84.1	Panoptic Segmentation, Self Attention	2020
Improving Semantic Segmentation via Decoupled Body and Edge Supervision [110]	83.7	Decoupled Multi-scale Feature Training	2020
Improving Semantic Segmentation via Video Propagation and Label Relaxation [111]	83.5	Joint Future Frame/Label Propagation, Data Augmentation	2019
Hard Pixel Mining for Depth Privileged Semantic Segmentation [112]	83.4	Depth Information, Depth-Aware Loss	2019
Global Aggregation then Local Distribution in Fully Convolutional Networks [113]	83.3	Global Aggregation, Local Distribution	2019

Table 2: Cityscapes Pixel-Level Semantic Labeling Task Top Performing Models

shows, real-time semantic segmentation is a reality and several architectures achieve accuracy close to state-of-the-art semantic segmentation models.

8 Summary

This work provides an extensive summary of the most recent advances in semantic image segmentation with deep learning methods focusing on real-time applications. It starts with an explanation of the segmentation task and how it differs from similar tasks, continues with a history of early segmentation methods, and provides a detailed description of the different deep learning approaches of the last decade. An extensive list of techniques to improve the efficiency of deep learning networks, by optimizing different aspects of the network, is then provided and the trade-offs in these design choices are explained. The most widely used benchmark data sets are subsequently described, followed by a list of metrics used to evaluate the accuracy and efficiency of the proposed models. Finally, performance tables are provided to summarize the state-of-the-art approaches in the area of semantic segmentation, both from an accuracy perspective, as well as an efficiency one.

Model	FPS	mIoU	Methods	Year
FastSCNN (quarter-resolution) [85]	485	51.9	Two-branch Networks, Depthwise Separable Convolutions	2019
FastSCNN (half-resolution) [85]	286	63.8	Two-branch Networks, Depthwise Separable Convolutions	2019
FasterSeg [114]	163	71.5	Neural Architecture Search	2020
LiteSeg [115]	161	67.8	Depthwise Separable Convolutions, Dilated Convolutions	2019
Partial Order Pruning [116]	143	71.4	Neural Architecture Search, Pruning	2019
RPNNet [117]	125	68.3	Early Downsampling, Residual Blocks	2019
FastSCNN [85]	123	68	Two-branch Networks, Depthwise Separable Convolutions	2019
Spatial Sampling Network for Fast Scene Understanding [118]	113	68.9	Smaller Decoder Size	2019
ESPNet [80]	112	60.3	Dilated Convolutions, Depthwise Separable Convolutions	2018
Efficient Dense Modules of Asymmetric Convolution [119]	108	67.3	Dilated Convolutions, Asymmetric Convolutions	2018

Table 3: Cityscapes Pixel-Level Semantic Labeling Task Top Performing Real-Time Models

Recent advances in deep learning methods, contemporaneously with a rapid increase in image capturing capabilities, have made image segmentation a crucial tool in a plethora of applications, from medical imaging to time-critical applications such as autonomous driving. This survey summarizes recent breakthroughs that transformed the field of image segmentation and provides a comprehensive insight on the design choices that led to this transformation.

References

- [1] Krizhevsky, Alex, Sutskever, Ilya, Hinton, Geoffrey. (2012). "ImageNet Classification with Deep Convolutional Neural Networks", Neural Information Processing Systems.
- [2] Z. Zhao, P. Zheng, S. Xu, X. Wu, "Object Detection with Deep Learning: A Review," arXiv:1807.05511v2.
- [3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL

- Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, 2010.
- [4] A. Novikov, D. Lenis, D. Major, J. Hladůvka, M. Wimmer, K. Bühler "Fully Convolutional Architectures for Multi-Class Segmentation in Chest Radiographs," *arXiv:1701.08816*.
 - [5] Wang G., Li W., Ourselin S., Vercauteren T. (2019) Automatic Brain Tumor Segmentation Using Convolutional Neural Networks with Test-Time Augmentation. In: Crimi A., Bakas S., Kuijf H., Keyvan F., Reyes M., van Walsum T. (eds) *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. BrainLes 2018. Lecture Notes in Computer Science*, vol 11384. Springer, Cham.
 - [6] V. Mukha, I. Sharony, "Disparity Image Segmentation For ADAS," *arXiv:1806.10350*.
 - [7] A. Sagar, R. Soundrapandiyar, "Semantic Segmentation With Multi Scale Spatial Attention For Self Driving Cars," *arXiv:2007.12685*.
 - [8] Yoshihara, A., Hascoet, T., Takiguchi, T., Ariki, Y., "Satellite Image Semantic Segmentation Using Fully Convolutional Network" (2018).
 - [9] A. King, S. M. Bhandarkar and B. M. Hopkinson, "A Comparison of Deep Learning Methods for Semantic Segmentation of Coral Reef Survey Images," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, 2018, pp. 1475-14758.
 - [10] Andres Milioto, Philipp Lottes, Cyrill Stachniss, "Real-time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs," *arXiv:1709.06764*.
 - [11] J. Martinsson and O. Mogren, "Semantic Segmentation of Fashion Images Using Feature Pyramid Networks," 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Korea (South), 2019, pp. 3133-3136.
 - [12] L. S. Davis, A. Rosenfeld and J. S. Weszka, "Region Extraction by Averaging and Thresholding," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-5, no. 3, pp. 383-388, May 1975, doi: 10.1109/TSMC.1975.5408419.
 - [13] Salem Saleh Al-amri, N.V. Kalyankar and Khamitkar S.D., "Image Segmentation by Using Threshold Techniques," *arXiv:1005.4020*.
 - [14] M. Özden and E. Polat, "Image segmentation using color and texture features," 2005 13th European Signal Processing Conference, Antalya, 2005, pp. 1-4.
 - [15] H. P. Ng, S. H. Ong, K. W. C. Foong, P. S. Goh and W. L. Nowinski, "Medical Image Segmentation Using K-Means Clustering and Improved Watershed Algorithm," 2006 IEEE Southwest Symposium on Image Analysis and Interpretation, Denver, CO, 2006, pp. 61-65.

- [16] Z. Huang and D. Liu, "Segmentation of Color Image Using EM algorithm in HSV Color Space," 2007 International Conference on Information Acquisition, Seogwipo-si, 2007, pp. 316-319.
- [17] W. Tao, H. Jin and Y. Zhang, "Color Image Segmentation Based on Mean Shift and Normalized Cuts," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 37, no. 5, pp. 1382-1389, Oct. 2007.
- [18] S. N. Sulaiman and N. A. Mat Isa, "Adaptive fuzzy-K-means clustering algorithm for image segmentation," in IEEE Transactions on Consumer Electronics, vol. 56, no. 4, pp. 2661-2668, November 2010.
- [19] N. Senthilkumaran and R. Rajesh, "Edge Detection Techniques for Image Segmentation – A Survey of Soft Computing Approaches," International Journal of Recent Trends in Engineering, Vol. 1, No. 2, May 2009.
- [20] Roberts, Lawrence. (1963). Machine Perception of Three-Dimensional Solids.
- [21] Sobel, Irwin. (2014). An Isotropic 3x3 Image Gradient Operator. Presentation at Stanford A.I. Project 1968.
- [22] Prewitt, J.M.S. (1970). "Object Enhancement and Extraction". Picture processing and Psychopictorics. Academic Press.
- [23] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, Aug. 2000.
- [24] Peng B., Zhang L., Yang J. (2010) "Iterated Graph Cuts for Image Segmentation". In: Zha H., Taniguchi R., Maybank S. (eds) Computer Vision – ACCV 2009. ACCV 2009. Lecture Notes in Computer Science, vol 5995. Springer, Berlin, Heidelberg.
- [25] C. L. Zhao, "Image segmentation based on fast normalized cut." Open Cybernetics and Systemics Journal, 2015, 9(1):28-31.
- [26] Nowozin, Sebastian, and Christoph H. Lampert. "Structured learning and prediction in computer vision." Foundations and Trends in Computer Graphics and Vision.
- [27] John Lafferty, Andrew McCallum, and Fernando C.N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," June 2001.
- [28] Philipp Krähenbühl, Vladlen Koltun, "Efficient Inference in Fully-Connected CRFs with Gaussian Edge Potentials," arXiv:1210.5644.
- [29] K Simonyan, A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556v6.
- [30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, "Going Deeper with Convolutions," arXiv:1409.4842v1.

- [31] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3431-3440, doi: 10.1109/CVPR.2015.7298965.
- [32] H. Noh, S. Hong and B. Han, "Learning Deconvolution Network for Semantic Segmentation," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1520-1528, doi: 10.1109/ICCV.2015.178.
- [33] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
- [34] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.
- [35] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs," arXiv:1412.7062v4.
- [36] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834-848, 1 April 2018, doi: 10.1109/TPAMI.2017.2699184.
- [37] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, Philip H. S. Torr, "Conditional Random Fields as Recurrent Neural Networks," arXiv:1502.03240v3.
- [38] Wei Liu, Andrew Rabinovich, Alexander C. Berg, "ParseNet: Looking Wider to See Better," arXiv:1506.04579v2.
- [39] Kim, Dong, Arsalan, Muhammad, Owais, Muhammad, Park, Kang. (2020). "ESSN: Enhanced Semantic Segmentation Network by Residual Concatenation of Feature Maps." IEEE Access. PP. 10.1109/ACCESS.2020.2969442.
- [40] Tao Yang, Yan Wu, Junqiao Zhao, Linting Guan, "Semantic Segmentation via Highly Fused Convolutional Network with Multiple Soft Cost Functions," arXiv:1801.01317v1.
- [41] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). "Generative Adversarial Networks." Proceedings of the International Conference on Neural Information Processing Systems (NIPS 2014). pp. 2672–2680.

- [42] Kevin Schawinski, Ce Zhang, Hantian Zhang, Lucas Fowler, Gokula Krishnan Santhanam, "Generative Adversarial Networks recover features in astrophysical images of galaxies beyond the deconvolution limit," arXiv:1702.00403v1.
- [43] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, Joshua B. Tenenbaum, "Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling," arXiv:1610.07584v2.
- [44] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," arXiv:1609.04802v5.
- [45] Pauline Luc, Camille Couprie, Soumith Chintala, Jakob Verbeek, "Semantic Segmentation using Adversarial Networks," NIPS Workshop on Adversarial Training, Dec 2016, Barcelona, Spain.
- [46] N. Souly, C. Spampinato and M. Shah, "Semi Supervised Semantic Segmentation Using Generative Adversarial Network," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017, pp. 5689-5697, doi: 10.1109/ICCV.2017.606.
- [47] X. Zhang, X. Zhu, 3. X. Zhang, N. Zhang, P. Li and L. Wang, "SegGAN: Semantic Segmentation with Generative Adversarial Network," 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM), Xi'an, 2018, pp. 1-5, doi: 10.1109/BigMM.2018.8499105.
- [48] Deep Learning (Ian J. Goodfellow, Yoshua Bengio and Aaron Courville), MIT Press, 2016.
- [49] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, Aaron Courville, "ReSeg: A Recurrent Neural Network-Based Model for Semantic Segmentation," 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, 2016, pp. 426-433, doi: 10.1109/CVPRW.2016.60.
- [50] Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron Courville, Yoshua Bengio, "ReNet: A Recurrent Neural Network Based Alternative to Convolutional Networks," arXiv:1505.00393v3.
- [51] Andreas Pfeuffer, Karina Schulz, Klaus Dietmayer, "Semantic Segmentation of Video Sequences with Convolutional LSTMs," arXiv:1905.01058v1.
- [52] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv:1409.0473v7.

- [53] L. Chen, Y. Yang, J. Wang, W. Xu and A. L. Yuille, "Attention to Scale: Scale-Aware Semantic Image Segmentation," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 3640-3649, doi: 10.1109/CVPR.2016.396.
- [54] Hanchao Li, Pengfei Xiong, Jie An, Lingxue Wang, "Pyramid Attention Network for Semantic Segmentation," arXiv:1805.10180v3.
- [55] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual Attention Network for Scene Segmentation," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 3141-3149, doi: 10.1109/CVPR.2019.00326.
- [56] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei and W. Liu, "CCNet: Criss-Cross Attention for Semantic Segmentation," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 603-612, doi: 10.1109/ICCV.2019.00069.
- [57] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, "PSANet: Point-wise Spatial Attention Network for Scene Parsing". In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11213.
- [58] C. Kaul, S. Manandhar and N. Pears, "Focusnet: An Attention-Based Fully Convolutional Network for Medical Image Segmentation," 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 2019, pp. 455-458, doi: 10.1109/ISBI.2019.8759477.
- [59] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, Piotr Dollár, "Panoptic Segmentation," arXiv:1801.00868v3.
- [60] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, Liang-Chieh Chen, "Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation," arXiv:1911.10194v3.
- [61] Rohit Mohan, Abhinav Valada, "EfficientPS: Efficient Panoptic Segmentation," arXiv:2004.02307v2.
- [62] <https://www.Cityscapes-dataset.com/benchmarks/>
- [63] <http://host.robots.ox.ac.uk/pascal/VOC/>
- [64] Andrew Tao, Karan Sapra, Bryan Catanzaro, "Hierarchical Multi-Scale Attention for Semantic Segmentation," arXiv:2005.10821v1.
- [65] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk, Quoc V. Le, "Rethinking Pre-training and Self-training," arXiv:2006.06882v1.
- [66] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," in IEEE Transactions on Information Theory, vol. 11, no. 3, pp. 363-371, July 1965.

- [67] Oppenheim, Alan V.; Schafer, Ronald W.; Buck, John R. (1999). Discrete-time signal processing (2nd ed.). Upper Saddle River, N.J.: Prentice Hall. ISBN 0-13-754920-2.
- [68] Michael Mathieu, Mikael Henaff, Yann LeCun, "Fast Training of Convolutional Networks through FFTs," arXiv:1312.5851v5.
- [69] Sheng Lin, Ning Liu, Mahdi Nazemi, Hongjia Li, Caiwen Ding, Yanzhi Wang, Massoud Pedram, "FFT-based deep learning deployment in embedded systems," 2018 Design, Automation and Test in Europe Conference and Exhibition (DATE), Dresden, 2018, pp. 1045-1050.
- [70] Song Han, Jeff Pool, John Tran, William J. Dally, "Learning both Weights and Connections for Efficient Neural Networks," arXiv:1506.02626v3.
- [71] Xinghao Chen, Yunhe Wang, Yiman Zhang, Peng Du, Chunjing Xu, Chang Xu, "Multi-Task Pruning for Semantic Segmentation Networks," arXiv:2007.08386v1.
- [72] Yihui He, Xiangyu Zhang, Jian Sun, "Channel Pruning for Accelerating Very Deep Neural Networks," arXiv:1707.06168v2.
- [73] J. Luo, H. Zhang, H. Zhou, C. Xie, J. Wu and W. Lin, "ThiNet: Pruning CNN Filters for a Thinner Net," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 10, pp. 2525-2538, 1 Oct. 2019.
- [74] Song Han, Huizi Mao, William J. Dally "Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding," arXiv:1510.00149v5.
- [75] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, Kwang-Ting Cheng, "Bi-Real Net: Enhancing the Performance of 1-bit CNNs With Improved Representational Capability and Advanced Training Algorithm," arXiv:1808.00278v5.
- [76] L. Sifre. Rigid-motion scattering for image classification. PhD thesis, Ph. D. thesis, 2014.
- [77] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017.
- [78] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv:1704.04861v1.
- [79] Fisher Yu, Vladlen Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," arXiv:1511.07122v3.
- [80] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, Hannaneh Hajishirzi, "ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation," arXiv:1803.06815v3.
- [81] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, Eugenio Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation," arXiv:1606.02147v1.

- [82] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 2818-2826.
- [83] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He, "Aggregated Residual Transformations for Deep Neural Networks," arXiv:1611.05431v2.
- [84] X. Zhang, X. Zhou, M. Lin and J. Sun, "ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 6848-6856.
- [85] Rudra P K Poudel, Stephan Liwicki, Roberto Cipolla, "Fast-SCNN: Fast Semantic Segmentation Network," arXiv:1902.04502v1.
- [86] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, Nong Sang, "BiSeNet V2: Bilateral Network with Guided Aggregation for Real-time Semantic Segmentation," arXiv:2004.02147v1.
- [87] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," arXiv:1502.03167v3.
- [88] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 648–656.
- [89] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014
- [90] <https://cocodataset.org/#home>
- [91] H. Caesar, J. Uijlings and V. Ferrari, "COCO-Stuff: Thing and Stuff Classes in Context," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 1209-1218.
- [92] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [93] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [94] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. Yuille. Joint object and part segmentation using deep learned potentials. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 1573–1581, 2015.

- [95] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In 2011 International Conference on Computer Vision, pages 991–998, 2011.
- [96] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, Antonio Torralba, "Semantic Understanding of Scenes through the ADE20K Dataset," arXiv:1608.05442v2.
- [97] <https://groups.csail.mit.edu/vision/datasets/ADE20K/>
- [98] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," arXiv:1604.01685v2.
- [99] <https://www.cityscapes-dataset.com/>
- [100] G. Ros, L. Sellart, J. Materzynska, D. Vazquez and A. M. Lopez, "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 3234-3243
- [101] <https://synthia-dataset.net/>
- [102] C. Liu, J. Yuen and A. Torralba, "Nonparametric scene parsing: Label transfer via dense scene alignment," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 1972-1979.
- [103] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. "LabelMe: a database and web-based tool for image annotation." IJCV, 77(1-3):157–173, 2008.
- [104] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. "Semantic object classes in video: A high-definition ground truth database." Pattern Recognition Letters, 30:88–97, 2009.
- [105] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.
- [106] <http://www.cvlibs.net/datasets/kitti/>
- [107] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D. Collins, Ekin D. Cubuk, Barret Zoph, Hartwig Adam, Jonathon Shlens, "Naive-Student: Leveraging Semi-Supervised Learning in Video Sequences for Urban Scene Segmentation," arXiv:2005.10266v4.
- [108] Yuhui Yuan, Xilin Chen, Jingdong Wang, "Object-Contextual Representations for Semantic Segmentation", arXiv:1909.11065v5.
- [109] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, Liang-Chieh Chen, "Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation," arXiv:2003.07853v2.

- [110] Xiangtai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, Yunhai Tong, "Improving Semantic Segmentation via Decoupled Body and Edge Supervision," arXiv:2007.10035v2.
- [111] Yi Zhu, Karan Sapra, Fitsum A. Reda, Kevin J. Shih, Shawn Newsam, Andrew Tao, Bryan Catanzaro, "Improving Semantic Segmentation via Video Propagation and Label Relaxation," arXiv:1812.01593v3.
- [112] Zhangxuan Gu, Li Niu, Haohua Zhao, Liqing Zhang, "Hard Pixel Mining for Depth Privileged Semantic Segmentation," arXiv:1906.11437v5.
- [113] Xiangtai Li, Li Zhang, Ansheng You, Maoke Yang, Kuiyuan Yang, Yunhai Tong, "Global Aggregation then Local Distribution in Fully Convolutional Networks," arXiv:1909.07229v1.
- [114] Wuyang Chen, Xinyu Gong, Xianming Liu, Qian Zhang, Yuan Li, Zhangyang Wang, "FasterSeg: Searching for Faster Real-time Semantic Segmentation," arXiv:1912.10917v2.
- [115] Taha Emara, Hossam E. Abd El Munim, Hazem M. Abbas, "LiteSeg: A Novel Lightweight ConvNet for Semantic Segmentation," arXiv:1912.06683v1.
- [116] Xin Li, Yiming Zhou, Zheng Pan, Jiashi Feng, "Partial Order Pruning: for Best Speed/Accuracy Trade-off in Neural Architecture Search," arXiv:1903.03777v2.
- [117] Xiaoyu Chen, Xiaotian Lou, Lianfa Bai, Jing Han, "Residual Pyramid Learning for Single-Shot Semantic Segmentation," arXiv:1903.09746v1.
- [118] Davide Mazzini, Raimondo Schettini, "Spatial Sampling Network for Fast Scene Understanding," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2019.
- [119] Shao-Yuan Lo, Hsueh-Ming Hang, Sheng-Wei Chan, Jing-Jhih Lin, "Efficient Dense Modules of Asymmetric Convolution for Real-Time Semantic Segmentation," arXiv:1809.06323v3.