electronics

MDPI

Review

Deep Learning-Based Frameworks for Semantic Segmentation of Road Scenes

Haneen Alokasi * and Muhammad Bilal Ahmad

Department of Computer Science, College of Computer Science and Information Technology, King Faisal University, Alahsa 31982, Saudi Arabia; mzulfiqar@kfu.edu.sa
* Correspondence: 218038142@student.kfu.edu.sa

Abstract: Semantic segmentation using machine learning and computer vision techniques is one of the most popular topics in autonomous driving-related research. With the revolution of deep learning, the need for more efficient and accurate segmentation systems has increased. This paper presents a detailed review of deep learning-based frameworks used for semantic segmentation of road scenes, highlighting their architectures and tasks. It also discusses well-known standard datasets that evaluate semantic segmentation systems in addition to new datasets in the field. To overcome a lack of enough data required for the training process, data augmentation techniques and their experimental results are reviewed. Moreover, domain adaptation methods that have been deployed to transfer knowledge between different domains in order to reduce the domain gap are presented. Finally, this paper provides quantitative analysis and performance evaluation and discusses the results of different frameworks on the reviewed datasets and highlights future research directions in the field of semantic segmentation using deep learning.

Keywords: deep learning; semantic segmentation; road scenes

Citation: Alokasi, H.; Ahmad, M.B. Deep Learning-Based Frameworks for Semantic Segmentation of Road Scenes. *Electronics* **2022**, *11*, 1884. https://doi.org/10.3390/electronics11121884

Academic Editors: Mannan Muhammad, Hoon-Seok Jang, Waqas ur Rahman and Hyunjin Park

Received: 25 April 2022
Accepted: 26 May 2022
Published: 15 June 2022

1. Introduction

Semantic segmentation is one of the most challenging tasks in the field of computer vision and is not an isolated task. Semantic segmentation means a complete scene understanding and is applied to images, videos, and 3D data. The basic idea behind semantic segmentation techniques is to segment an image into pixels and to assign a label to each pixel. The importance of scene understanding has increased due to the increased number of applications that depend on it. Examples of these applications are human–machine interaction [1], images search engines [2], and autonomous driving [3–5]. Any autonomous driving system must detect objects, segment the road, and recognize traffic signs.

The task of semantic segmentation is to generate inference from a coarse level to a fine level. Semantic segmentation uses classification techniques to predict each pixel of the entire input, then getting fine-grained inference by detection or localization in which the class labels and their spatial location information are provided. Semantic segmentation aims to have dense predictions that derive class labels for each pixel. As a result, every pixel will be labeled with its enclosing object's class. All of the instances of the same class are further separated and that is known as instance segmentation [6]. Moreover, the combination of semantic segmentation and instance segmentation is called panoptic segmentation, which means each pixel is assigned a class label and an instance label [7]. Figure 1 shows the evolution of semantic segmentation, instance segmentation, and panoptic segmentation.

Prior to 2000, multiple segmentation methods were proposed based on image processing such as clustering, texture features, region segmentation, and threshold segmentation [9,10]. In the past two decades, segmentation algorithms were categorized into four groups: classification [11], clustering [12], a combination of clustering and classification [13], and graph theory [14]. Since 2010, neural network models have improved,

and deep learning models based on segmentation algorithms have been developed [15]. Today, with the revolution of deep learning, most problems related to computer vision are addressed by using deep learning architectures. The most popular architecture is known as Convolutional Neural Network (CNN). CNN shows better efficiency and accuracy when compared to classical architectures. Examples of the early proposed CNN architectures based on segmentation algorithms include: a trained convolutional network for detecting, segmenting, and locating cells and nuclei in microscopic images [16], an approach that automatically segments the neural structures depicted in stacks of electron microscopy (EM) images [17], a method that uses a multiscale convolutional network trained from raw pixels for extracting dense feature vectors to encode multiple sizes centered regions on each pixel [18], a novel CNN architecture for simultaneous detection and segmentation (SDS) that detects all instances of an object in an image and marks the pixels that belong to each instance [19], and a geocentric embedding for learning feature representations with CNNs, in addition to a decision forest approach that classifies pixels as foreground or background [20].

(a) Image (b) Semantic Segmentation (c) Instance Segmentation (d) Panoptic Segmentation

Figure 1. The results of semantic segmentation, instance segmentation, and panoptic segmentation by Panoptic-Deeplab [8] on Cityscapes images [5].

However, the most used Deep Convolutional Neural Networks (DCNNs) by the deep learning community for the task of semantic segmentation are Fully Convolutional Networks (FCNs) [21]. The pipeline of FCN extends the basic CNN architecture. Furthermore, FCN only has convolutional and pooling layers, which can predict any arbitrary-sized input image, unlike CNN, which has fixed fully connected layers that predict labels only for particular sizes of the input images. Generally, the FCN prediction results have low resolution due to the downsampled feature maps through multiple alternated convolutional and pooling layers. To deal with this problem, various architectures have been proposed such as SegNet [22], which has a decoder that upsamples its lower resolution input feature maps, and UNet [23], which has a massive number of feature channels in the upsampling part that propagate context information to higher resolution layers. However, different deep network architectures have been used over the past several years and have become widely known standards for semantic segmentation. Examples include VGG-16 [24], GoogLeNet [25], ResNet [26], DenseNet [27], AlexNet [28], and HANet [29]. Moreover, some of these architectures are in use as backbone networks for the recently proposed deep networks. Today, new architectures have been proposed to address not only semantic segmentation problems but also 3D semantic segmentation and real-time semantic segmentation problems.

The rest of this paper is organized as follows. Section 2 describes well-known and new datasets in the field that are used for semantic segmentation. Generally, data provided are not enough for semantic segmentation, Section 3 discusses data augmentation techniques used to increase data, the disadvantages of those techniques, and experimental results. Furthermore, Section 4 presents domain adaptation methods that have been deployed to transfer knowledge between different domains in order to reduce the domain gap. Section 5 provides a comprehensive overview of state-of-the-art semantic segmentation frameworks, organized in chronological order. Those frameworks were selected based on their promised performance. Then, Section 6 discusses some popular evaluation metrics used for measuring the performance of semantic segmentation systems along with the numeric results of the reviewed frameworks on some standard datasets mentioned in Section 2. The results are divided into three groups based on the task of the frameworks:

semantic segmentation, 3D semantic segmentation, or real-time semantic segmentation. Finally, Section 7 highlights some future research directions on the field, while Section 8 concludes the paper.

2. Datasets

Over the past few years, with the improvement of deep learning techniques, numerous datasets have been created for semantic segmentation tasks. This section describes new and widely known datasets that are commonly used for semantic segmentation. Table 1 provides some useful information for all of the described datasets, such as their class number, data format, and training/validation/testing splits. Figure 2 shows sample images from most of the discussed datasets.

Table 1. An overview of the datasets reviewed in this paper organized in chronological order.

Dataset Name	Purpose	Number of Classes	Resolution	Real/Synthetic	Training	Validation	Test
CamVid [30]	Urban (Driving)	32	960 × 720	Real	701	N/A	N/A
CamVid-Sturgess [31]	Urban (Driving)	11	960 × 720	Real	367	100	233
KITTI-Layout [32]	Urban/Driving	3	Variable	Real	323	N/A	N/A
Microsoft COCO [33]	Generic	>80	Variable	Real	82,783	40,504	81,434
PASCAL VOC 2012 [34]	Generic	21	Variable	Real	1464	1449	Private
KITTI-Ros [35]	Urban/Driving	11	Variable	Real	170	N/A	46
KITTI-Zhang [36]	Urban/Driving	10	1226 × 370	Real	140	N/A	112
Cityscapes [5] (fine)	Urban	30 (8)	2048 × 1024	Real	2975	500	1525
Cityscapes [5] (coarse)	Urban	30 (8)	2048 × 1024	Real	22,973	500	N/A
SYNTHIA [37]	Urban/Driving	13	960 × 720	Synthetic	13,407	N/A	N/A
GTA5 [38]	Driving	19	1914 × 1052	Synthetic	N/A	N/A	N/A
Mapillary Vistas [39]	Urban	66	High	Real	18,000	2000	5000
ADE20K [40]	Urban/Indoor	150	High	Real	27,574	N/A	2000
SemanticKITTI [41]	Driving	28	High	Real	23,201	N/A	20,351
nuScenes [42]	Driving	23	High	Real	700	150	150
Apolloscape [43]	Driving	36	3384 × 2710	Real	N/A	N/A	N/A

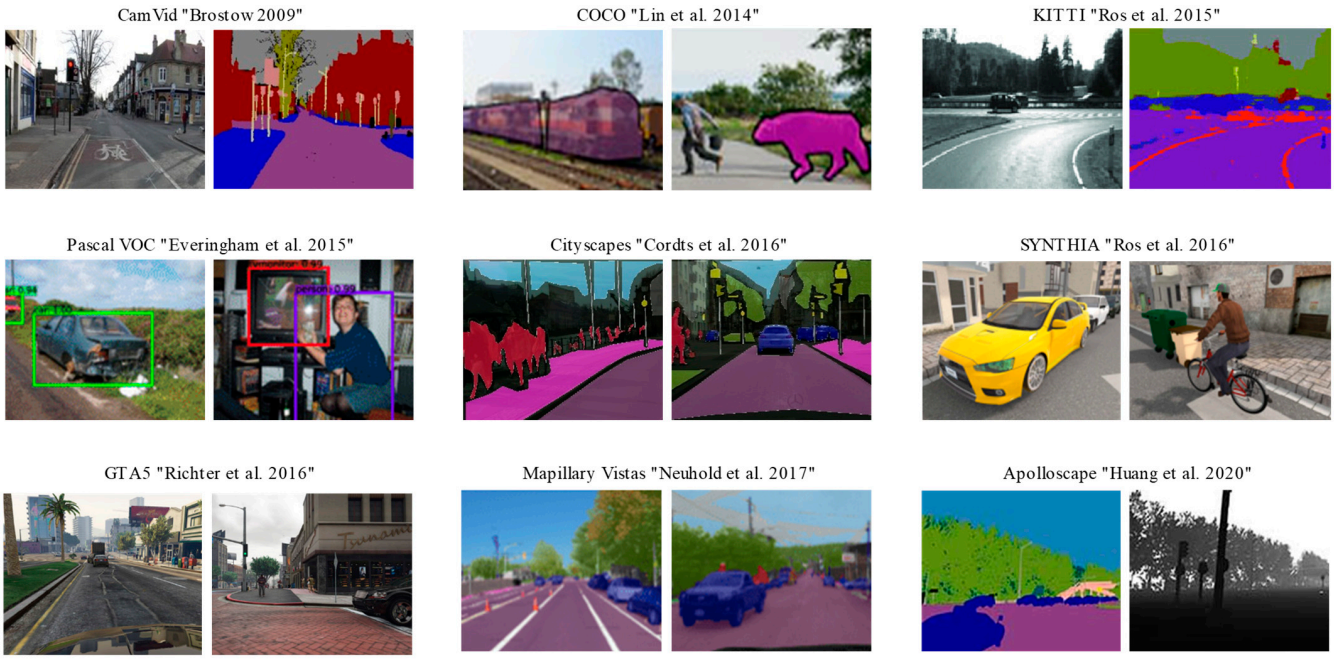


Figure 2. Sample images from some of the reviewed semantic segmentation datasets. Brostow 2009 [30]; Lin et al. 2014 [33]; Ros et al. 2015 [35]; Everingham et al. 2015 [34]; Cordts et al. 2016 [5]; Ros et al. 2016 [37]; Richter et al. 2016 [38]; Neuhold et al. 2017 [39]; Huang et al. 2020 [43].

2.1. Cambridge-Driving Labeled Video Database (CamVid)

CamVid (http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/, accessed 16 May 2022) [30,44] is considered to be the first set of videos with object class semantic labels. Data were captured from a driving car perspective, where a camera was attached to the dashboard of a car. CamVid provides ground truth labels that associate every pixel with a class from 32 classes: sky, void, bridge, wall, building, fence, tree, sidewalk, vegetation, animal, pedestrian, child, bicyclist, cart luggage, traffic, sign, lane markings (driving, nondriving), traffic light, traffic cone, pole, miscellaneous text, parking block, tunnel, road, road shoulder, archway, car, train, motorcycle, bus/truck, truck/SUV/pickup, and other moving objects. However, Sturgess et al. [31] introduced the partition that distributed the dataset into 367 training images, 100 validation images, and 233 testing images.

2.2. KITTI

KITTI (http://www.cvlibs.net/datasets/kitti/, accessed 16 May 2022) [45] is one of the most popular datasets being used for autonomous driving. Furthermore, hours of traffic situations have been recorded in KITTI with different sensor modalities, containing a 3D laser scanner, high-resolution RGB, and grayscale stereo cameras. Unlike other datasets, KITTI does not include ground truth for semantic segmentation. However, researchers have made manual annotation on parts of the dataset to suit their needs. Zhang et al. [36] annotated 140 training images and 112 testing images for 10 classes: car, road, sidewalk, fence, cyclist, pedestrian, building, sky, pole/sign, and vegetation. Alvarez et al. [32] and Ros and Alvarez [46] generated the ground truth from the road detection challenge for 323 images with three classes: sky, vertical, and road. Ros et al. [35] labeled 216 images (170 for training and 46 for testing) with 11 classes: road, sign, sidewalk, pole, fence, building, tree, car, sky, bicyclist, and pedestrian.

2.3. Microsoft Common Objects in Context (COCO)

The COCO (https://cocodataset.org/#home, accessed 16 May 2022) [33] challenge contains a large-scale dataset of images for recognition, segmentation, and captioning. It has more than 80 classes and consists of more than 82,783 training images, 40,504 validation images, and 81,434 testing images. Specifically, the test set is divided into four groups: test-dev contains 20,000 images for additional validation or debugging, test-standard contains 20,000 images as the default testing set for the challenge and for a comparison between state-of-the-art techniques, test-challenge contains 20,000 images used in the challenge for the submitted methods to the evaluation server, and test-reverse contains 20,000 images used in the challenge for preventing any possible overfitting. The main reason for the dataset's popularity is its large-scale.

2.4. Pascal Visual Object Classes (VOC)

The PASCAL VOC (http://host.robots.ox.ac.uk/pascal/VOC/, accessed 16 May 2022) [34] challenge contains a dataset of ground-truth annotated images and has five contests: classification, detection, segmentation, action classification, and person layout. The dataset is grouped into categories such as households, vehicles, animals, and has 21 classes: sofa, table, chair, potted plant, dining, TV/monitor, bottle, car, train, bus, boat, motorbike, bicycle, airplane, sheep, horse, dog, cat, bird, cow, and person. Moreover, if the pixel does not belong to any of the classes, then it is considered as background. However, the dataset has 1464 training images, 1449 validation images, and a private testing set. PASCAL VOC is considered to be the most well-known dataset for semantic segmentation tasks. In literature, almost all of the outstanding methods have been submitted to PASCAL VOC's performance evaluation server for validation against the private testing set.

2.5. Cityscapes

Cityscapes (https://www.cityscapes-dataset.com/, accessed 14 April 2022) [5] is a massive urban scenes dataset that focuses on semantic understanding. The data were

captured in 50 cities during different months, weather conditions, and times of day. The data were initially recorded as a video, then the frames were selected manually to collect features such as scene layout, dynamic objects, and backgrounds. Cityscapes provides annotations for 30 classes categorized into 8 groups: sky, nature, vehicle, surface, construction, human, object, and void. The dataset contains 2000 coarse annotated images and 5000 fine annotated images.

2.6. SYNTHetic Collection of Imagery and Annotations (SYNTHIA)

SYNTHIA (https://synthia-dataset.net/, accessed 16 May 2022) [37] is a massive dataset that contains realistic photo renderings of a virtual city. The dataset has 13,407 training images and is used mostly for scene understanding, especially in autonomous driving scenarios. Fine-grained, pixel-level annotations were provided by the dataset for 13 classes: car, road, fence, sidewalk, pole, building, sign, vegetation, sky, misc, land-marking, cyclist, and pedestrian. The dataset is also grouped by scene's variety: lighting conditions and weather, seasons, and dynamic objects.

2.7. Grand Theft Auto 5 (GTA5)

The GTA5 (https://download.visinf.tu-darmstadt.de/data/from_games/, accessed 16 May 2022) [38] dataset contains 24,966 semantically labeled synthetic images extracted from the Grand Theft Auto 5 video game. The images are all from a car perspective in the roads of American-style virtual cities [38]. The dataset has 19 classes: road, building, sky, sidewalk, vegetation, car, terrain, wall, truck, pole, fence, bus, person, traffic light, traffic sign, train, motorcycle, rider, and bicycle.

2.8. Mapillary Vistas

Mapillary Vistas (https://www.mapillary.com/dataset/vistas, accessed 16 May 2022) [39] is one of the largest-scale, street-level image datasets. It contains 25,000 high-resolution images divided into 18,000 images for the training set, 2000 for the validation set, and 5000 for the testing set. These images have been annotated into 66 object groups in addition to instance-specific labels for 37 classes. Moreover, the annotation was performed by using a polygon for delineating individual objects in a dense and fine-grained style. A comparison of the total amount of fine annotation for this dataset to the Cityscape dataset shows that it is five times larger and contains images that have been captured at different times of day and in different weather and season conditions. The images were taken by experienced photographers with different imaging devices, such as action cameras, tablets, mobile phones, and professional capturing rigs. Mapillary Vistas was designed and compiled to cover a variety of details and geographic extent.

2.9. ADE20K

ADE20K (http://groups.csail.mit.edu/vision/datasets/ADE20K/, accessed 16 May 2022) [40] is a fully annotated image dataset. All images in the dataset are annotated with objects, and many objects are annotated with their parts. Moreover, there is additional information about each object, such as whether it is cropped or occluded. The current version of ADE20K dataset has 27,574 training images and 2000 testing images. Furthermore, the validation images are exhaustively annotated with parts, and the parts are not exhaustively annotated with the training images. Figure 3 shows sample images and annotations.

2.10. SemanticKITTI

SemanticKITTI (http://www.semantic-kitti.org/, accessed 16 May 2022) [41] is a massive dataset for semantic scene understanding and is based on the KITTI Vision benchmark. All sequences of the KITTI dataset have been annotated, and automotive Light Detection and Ranging (LiDAR) has been employed. For the complete 360-degree field of view, dense pointwise annotations have been provided. The SemanticKITTI dataset contains 28 classes, including moving and nonmoving traffic objects with distinct classes such as cars, mo-

torcycles, trucks, bicyclists, and pedestrians. The original split of the 22 sequences from the KITTI dataset was adopted for the SemanticKITTI dataset, where 00 to 10 sequences represent the training set, and 11 to 21 sequences represent the test set. Furthermore, 23,201 full 3D scans have been provided for the training set and 20,351 for the testing set.

Figure 3. Images from ADE20K dataset densely annotated in detail with objects and their parts. The top row shows sample input images, the middle row shows the annotations of objects, and the bottom row shows the annotations of objects' parts [40].

2.11. nuScenes

The nuScenes (https://www.nuscenes.org/, accessed 16 May 2022) [42] is a large-scale dataset developed by Aptiv Autonomous Mobility. nuScenes dataset is publicly available for supporting research in the computer vision field specifically for autonomous driving. This dataset is inspired by the KITTI dataset, but compared to KITTI, nuScenes contains 100 times as many images and 7 times as many objects' annotations. The data were collected in Singapore and Boston, both major cities known for heavy traffic and challenging driving situations. The dataset also contains 1000 scenes, each 20 s long. It shows a different set of unexpected behaviors, traffic situations, and driving maneuvers. These scenes are fully annotated with 3D bounding boxes for 23 classes and 8 attributes. The complexity of the nuScenes dataset encourages the development of safe driving methods. However, nuScenes is the first dataset that carries a full autonomous car sensor suite: six cameras, five radars, and one LiDAR, all with a 360-degree field of view.

2.12. ApolloScape

ApolloScape (http://apolloscape.auto/, accessed 16 May 2022) [43] contains large and rich labeling that includes landmark labeling, stereo, per-pixel semantic labeling, instance segmentation, holistic semantic dense point cloud for each location, high accurate site for each frame in different driving videos from various cities and daytime, and 3D car instance. ApolloScape dataset has 36 classes and 147,000 images with the corresponding pixel-level annotation for each image. The dataset includes depth maps and poses information for a static background. Riegl VMX-1HA, which has a VMX-CS6 camera system, was used

to capture all of the images in the dataset. Like the Cityscape dataset, ApolloScape has similar specification classes but also has a tricycle class that includes all types of three-wheeled vehicles.

3. Data Augmentation

Data augmentation is a process of expanding a dataset by applying various techniques. Wong et al. [47] presented data augmentation as a reliable way to improve the performance of machine learning algorithms, specifically deep architectures, by generating more samples to expand the dataset, avoiding overfitting, and increasing generalization capabilities. Augmentation techniques can be based on image manipulations, such as geometric transformations, color space transformation, random erasing, and kernel filters, or can be based on deep learning, such as Generative Adversarial Networks (GANs) [48] based augmentation, feature space augmentation, and neural style transfer. This section explains different augmentation techniques, discusses their disadvantages, and reports experimental results.

3.1. Data Augmentation Techniques Based on Image Manipulation

Most popular augmentations are based on geometric transformations such as cropping, rotation, color space, flipping, noise injection, and translation. These augmentations may or may not preserve the label post-transformation, depending on the applied augmentation and the data. In some cases, that means the model's ability to predict the output might be decreased after the augmentation. To solve such a problem, the labels post-transformation must be refined [49], which is a computationally expensive process.

Furthermore, geometric transformations are easily implemented and considered to be good solutions for positional bias problems that might occur in the training data. However, there are some disadvantages of geometric transformations, such as requiring longer training time, extra computation costs, and additional memory. Moreover, some of the geometric transformation techniques, such as random cropping and translation, require manual observation to be sure that the image label is not altered. In some real-world applications, such as medical image analysis, there is a need for more complex biases distancing than translation and positional variances. Therefore, the use of geometric transformations is relatively limited.

Another transformation based on image manipulation known as color space transformation or photometric transformation is used to alter the color distribution on images. Moreover, important color information can be discarded by color space transformation. For various tasks, colors can be major distinctive features. In some situations, this transformation is considered to be a non-label preserving transformation. However, color biases can be eliminated when color space transformation is applied. Color space transformation and geometric transformation have the same disadvantages, which are an increase of training time, computation costs, and memory. Taylor and Nitschke [50] presented a comparative study on the effectiveness of color space and geometric transformations. They studied flipping, $-30°$ to $30°$ rotations, and cropping as geometric transformations, in addition to color jittering, edge enhancement, and Fancy Principal Component Analysis (PCA) as color space transformations. Taylor and Nitschke [50] implemented those augmentations on the Caltech101 [51] dataset filtered to 8421 images of size 256×256. Their results are shown in Table 2. Cropping geometric transformation has the most accurate classifier, based on the results of Taylor and Nitschke [50].

Random erasing [52] is another data augmentation technique that is based on image manipulation. The technique was inspired by dropout regularization mechanisms. Particularly, random erasing was designed for compacting image recognition challenges because of occlusion. Occlusion means when some of an object's parts in an image are unclear. To overcome occlusion, the random erasing technique forces the model to learn more descriptive features about the image. In this way, the model will prevent overfitting to a particular feature of the image. The random erasing technique can guarantee that a network

is paying attention to the entire image, not just to a part of it. Zhong et al. [52] evaluated random erasing on medium-scale and large-scale datasets. Their results showed that the models trained with random erasing have remarkable improvement. Table 3 reports their results on ImageNet-2012 [53] validation set with three different architectures, which are ResNet-34, ResNet-50, and ResNet-101. A disadvantage to this technique, however, is that it is not usually considered as a label-preserving transformation.

Table 2. Taylor and Nitschke's experiments' results on Caltech 101 dataset [50].

	Top-1 Accuracy	Top-5 Accuracy
Baseline	48.13 ± 0.42%	64.50 ± 0.65%
Flipping	49.73 ± 1.13%	67.36 ± 1.38%
Rotating	50.80 ± 0.63%	69.41 ± 0.48%
Cropping	**61.95 ± 1.01%**	**79.10 ± 0.80%**
Color Jittering	**49.57 ± 0.53%**	67.18 ± 0.42%
Edge Enhancement	49.29 ± 1.16%	66.49 ± 0.84%
Fancy PCA	49.41 ± 0.84%	**67.54 ± 1.01%**

The bold values indicate high performance.

Table 3. Test error (%) on ImageNet-2012 validation set [52].

Model	Baseline		Random Erasing	
	Top-1	Top-5	Top-1	Top-5
ResNet-34	25.22	8.01	24.89	7.71
ResNet-50	23.39	6.89	22.75	6.69
ResNet-101	20.98	5.73	20.43	5.30

Kernel filters are another popular data augmentation technique. Some classical ways to apply kernel filters are by sharpening and blurring images. Those filters work by sliding an $n \times n$ matrix across an image with either a high contrast horizontal or vertical edge filter to get a sharper image or using a Gaussian blur filter to get a blurrier image. Sharpening an image might encapsulate additional details about important objects in the image. Additionally, blurring an image might increase the motion blur resistance during testing. Kang et al. [54] presented a unique kernel filter technique known as PatchShuffle regularization. It randomly swaps the pixels value into an $n \times n$ sliding window. The efficiency of Kang et al.'s [54] technique has been proven through their experiments on Canadian Institute for Advanced Research 10 classes (CIFAR-10) dataset [55], where a 5.66% error rate was achieved with the use of PatchShuffle regularization technique, compared to a 6.33% error rate when the technique was not used. However, a disadvantage of kernel filters is that they are very similar to the internal mechanisms of CNNs. Specifically, parameter kernels in CNNs learn the best way of representing images layer by layer. Hence, kernel filters can be implemented in a better way as a layer of the CNN instead of as a data augmentation technique applied to the dataset.

3.2. Data Augmentation Techniques Based on Deep Learning

The augmentation techniques discussed previously, which are based on image manipulation, are applied to images in the input space, unlike neural networks, which are extremely powerful at mapping high-dimensional inputs into lower-dimensional representations. Furthermore, neural networks have a sequential process, which can be manipulated to separate the intermediate representations from the network. In a fully connected layers network, the lower-dimensional representations of images can be extracted and isolated. The lower-dimensional representations found in high-level layers of a CNN are known as the feature space. DeVries and Taylor [56] discussed using the feature space technique for data augmentation. They also proposed adding noise, extrapolating, and interpolating as general forms of feature space augmentation.

To perform feature space augmentation on images, the use of autoencoders is beneficial. Autoencoders work by dividing the network into halves, where the first half is the encoder and the second half is the decoder. The encoder maps images into low-dimensional vector representations in a way in which the decoder can reconstruct the vectors back into the original images. For feature space augmentation, the encoded representation is used. If necessary, feature space augmentation can be implemented with autoencoders to reconstruct the new vectors back into input space. It is also possible to isolate vector representations from a CNN to implement only feature space augmentation. Furthermore, to generate new samples, DeVries and Taylor [56] applied interpolating or extrapolating in their classification experiments. They found that the classification accuracy is improved when datasets are augmented by extrapolating within a learned feature space, compared to no data augmentation. Their results on MNIST [57] and CIFAR-10 datasets averaged over 10 runs are shown in Table 4. However, the disadvantage of feature space augmentation is the difficulty of interpreting the vector data. Using autoencoders to recover the new vectors requires copying the entire encoding part of the CNN being trained. DCNNs require massive autoencoders, which are very difficult and consume time for training. Finally, Wong et al. [47] discovered that if transforming images in the data space is possible, data space augmentation will surpass feature space augmentation.

Table 4. Test error (%) on MNIST and CIFAR-10 datasets [56].

Model	MNIST	CIFAR-10
Baseline	1.093 ± 0.057	30.65 ± 0.27
Baseline + input space affine transformation	1.477 ± 0.068	-
Baseline + input space extrapolation	1.010 ± 0.065	-
Baseline + feature space extrapolation	$\mathbf{0.950 \pm 0.036}$	$\mathbf{29.24 \pm 0.27}$

Another great data augmentation tool based on deep learning is known as a neural style transfer [58]. The basic idea of neural style transfer is manipulating the sequential representations of images across a CNN, where the style of an image can be changed to a different one while keeping its original content. Moreover, choosing specific styles to transfer images into for some applications, such as autonomous driving, can be obvious. For example, the training images can be transferred into rainy-to-sunny, winter-to-summer, or night-to-day scale. On the other hand, in some applications, it might be difficult to choose the styles into which to transfer images. However, the effort required to select styles to transfer into is considered a disadvantage of neural style transfer augmentation. Another disadvantage is that the problem of biases could be introduced into the dataset if the size of the style set is too small. The original technique [58] is not practical for data augmentation since its running time is very slow. An improvement to the original neural style transfer technique known as fast style transfer has been proposed by Johnson et al. [59]. The improvement includes extending the loss function from a per-pixel loss to a perceptual loss and uses a feed-forward network to stylize images. This improvement allows style transfer to run in a much faster way, but this limits transfer to a set of styles that is pretrained.

In deep learning applications, most datasets have a common problem: some classes contain a higher number of samples in the training set than other classes. For example, when building a model that detects rare samples, the number of images of rare samples will be very small compared to the other sample images. This difference in the dataset is known as a class imbalance. Generally, the learning algorithms are biased toward the majority classes with imbalanced datasets. As a result, for the minority class, there is a high misclassification. Furthermore, there are various methods that can reduce the gap in imbalanced datasets by generating synthetic data. For example, Deep Convolutional Generative Adversarial Networks (DCGANs) [60] can improve the classification performance by generating synthetic samples for a minority class. Tanaka and Aranha [61] proposed using GANs for data augmentation, where GANs have been used to generate synthetic data for a binary classification task of cancer detection. In addition, a decision tree classifier

remarkably achieved better performance when trained on the newly generated synthetic dataset than when trained on the original dataset. In some cases, class information must be included in the GAN model. This can be done by using conditional GAN, where the class information is fed to the generator. Three of the most popular conditional GAN architectures in the past couple of years are discussed as follows.

- Auxiliary Classifier GAN (ACGAN) [62]: In ACGAN, the discriminator classifies and discriminates between real and synthetic generated data, where a binary cross-entropy is included in the loss function for classification. In this way, the generator learns representative class samples and learns to generate more-realistic data. ACGAN methods are used to improve the training process of GANs.
- Data Augmentation GAN (DAGAN) [63]: DAGAN uses a lower-dimensional representation of real images to learn how to generate synthetic images. Figure 4 shows its architecture. The generator of DAGAN is composed of an encoder that takes a true image from class as input, then projects it down to a lower-dimensional manifold (bottleneck). Then, by transforming and concatenating a random vector with the bottleneck vector, these two vectors are passed to the decoder for generating an augmented image. Furthermore, the discriminator of DAGAN has been trained to discriminate between real images from the class and fake images that have been generated by the generator. However, the training process drives the network to generate new images from the existing ones that appear to be in the same class, whatever the class is, although the generated images look different enough to be different samples.

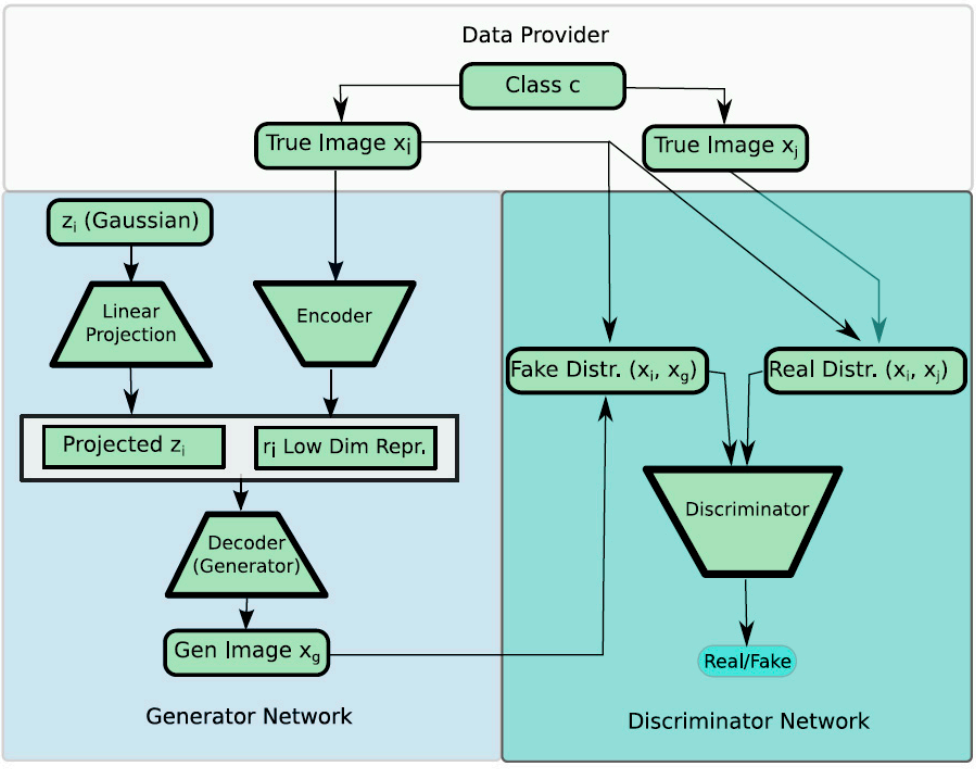
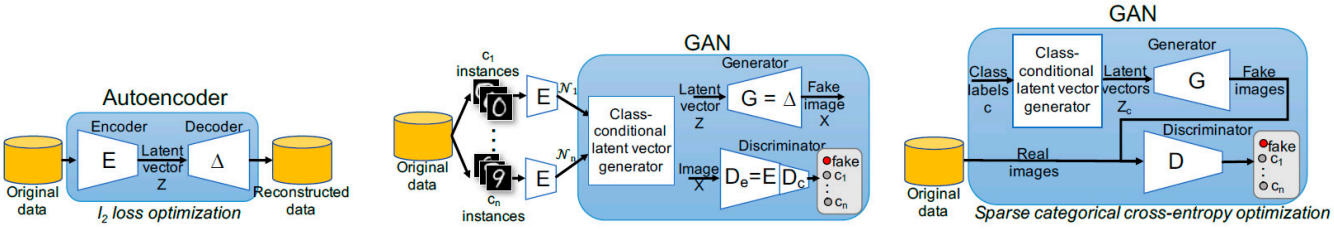


Figure 4. DAGAN architecture [63].

- Balancing GAN (BAGAN) [64]: It is an augmentation tool that restores balancing data in imbalanced datasets. Doing this was a challenge, considering the few minority class images that might not be enough for training a GAN. This problem was solved by including all available images of minority and majority classes during the training process. Figure 5 shows the three steps of the BAGAN training approach: autoencoder training, GAN initialization, and adversarial training. In this way, the model learns useful features from the majority classes and then uses these features to generate

images for minority classes. Additionally, to drive the generation process toward a target class, class conditioning has been applied in the latent space. An autoencoder is used for the generator, which helped in learning an accurate class conditioning in the latent space.



(a) Autoencoder training. (b) GAN initialization. (c) GAN training.

Figure 5. The three training steps of the BAGAN method [64].

However, comparing BAGAN with ACGAN, both were trained on the target datasets by using minority and majority classes. Furthermore, rather than transforming the existing data like in DAGAN, BAGAN generates new images of higher quality when trained with imbalanced datasets.

Most of the research papers that applied GANs to data augmentation were carried out in biomedical image analysis [65]. The results in these papers showed improvement in the classification boundaries derived from training with both real and generated data from GANs. Even though using GANs for data augmentation has great potential, it has disadvantages. For example, getting high-resolution outputs from the cutting-edge architecture of GAN is very difficult. Furthermore, increasing the output size of the images that the generator in a GAN produced can cause training instability. Another disadvantage of GANs is that a huge amount of data is required for training, which means GANs might not be a practical solution when the dataset is limited.

3.3. Comparing Data Augmentation Techniques

Data augmentation has multiple possibilities that can improve the performance of deep learning models. However, few comparative studies have been done on these augmentation techniques to try to show their performance differences. A study by Shijie et al. [66] explored the impact of various data augmentation techniques on image classification tasks with DCNNs. Shijie et al. [66] selected ImageNet and a subset of CIFAR-10 as the original dataset. The data augmentation techniques used on the experiment were GAN, Wasserstein GAN (WGAN), flipping, shifting, cropping, rotation, noise, color jittering, PCA jittering, and some combinations. The comparative study showed that on small-scale datasets under the same multiple increasing conditions, the performance evaluation was more obvious. Generally, cropping, flipping, WGAN, and rotation performed better than other techniques. Some combinations, such as flipping + cropping and flipping + WGAN, were the best overall and improved the classification performance on CIFAR-10 dataset by 3% and 3.5%, respectively.

4. Domain Adaptation

Training deep learning models relies on a large amount of pixel-level labeled data. Generally, data labeling is done manually, which is considered a difficult and time-consuming process. For the Cityscapes dataset, high-quality semantic labeling needs about 90 min per image [5]. Another way for training deep learning models is to use synthetic data (source-domain) adapted to real images (target-domain). Various studies have proposed domain adaptation methods to reduce the domain gap between labeled source-domain and unlabeled target-domain. Furthermore, domain adaptation can be divided into three categories based on the availability of labeled data in the target-domain, which are supervised, unsupervised, and semi-supervised. Tzeng et al. [67] addressed the supervised

domain adaptation by proposing a new CNN architecture which contains an adaptation layer and an additional domain confusion loss to learn a representation that is both domain-invariant and semantically meaningful. However, if the labeled data is not available in the target-domain, this is known as unsupervised domain adaptation. To deal with this situation, Tzeng et al. [68] proposed a framework that uses adversarial learning to improve the model's generalization ability. Further, when both labeled and unlabeled data are available in the target-domain, this is referred to as semi-supervised domain adaptation. Long et al. [69] introduced a deep adaptation network to address the semi-supervised domain adaptation. The hidden representations of all task-specific layers are embedded in a reproducing kernel Hilbert space, where the mean embedding of various domain distribution can be easily matched [69].

Most of the existing domain adaptation methods are proposed to address image classification tasks. There have been multiple methods introduced to address the semantic segmentation task [67–69]. Hoffman et al. [70] took this research area into another level with global and category specific adaptation methods that use pixel-level adversarial training. However, the discussed domain adaptation methods reduced the domain gap by mainly using adversarial training. Recently, some newly proposed methods have been considered the first attempts to introduce new models to domain adaptation for semantic segmentation. Some of these methods are discussed as follows.

- Self-Ensembling Attention Networks [71] is the first self-ensembling model introduced to domain adaptation. This model aims to improve the learning of domain-invariant features. Moreover, an attention mechanism is introduced into the proposed model to generate attention-aware features. This mechanism is used to calculate consistency loss in the target-domain. Moreover, the self-ensembling model has two major components: a student network that represents base networks and a teacher network that represents ensemble networks. The student network learns from the teacher network with the help of the consistency loss. As a result, the student network becomes more accurate and the teacher network gets closer to the correct labels in the target-domain. Hence, domain-invariant features can be learned correspondingly.
- Semantic-Edge Domain Adaptation [72] is the first attempt to use low-level edge information that can be easily adapted to guide the transfer of high-level semantic information. The semantic-edge domain adaptation model uses an edge stream for processing edge information to produce high-quality semantic boundaries over the target-domain. Moreover, an edge consistency loss is proposed to align the target semantic predictions with the generated semantic boundaries in addition to two entropy reweighting methods for enhancing the adaptation performance of the model.
- Self-Ensembling GAN (SE-GAN) [73] is a novel self-ensembling GAN for domain adaptation. SE-GAN adopted a self-ensembling model as the generator in the adversarial network to improve the adversarial training performance. SE-GAN has three major components: a student network, a teacher network, and a discriminator. Both the student and the teacher networks form a self-ensembling model that generates domain-invariant features. On the other hand, the discriminator determines whether the segmentation maps come from the source-domain or the target-domain. However, the teacher network in SE-GAN produces pseudo labels for the target-domain images and conducts self-training on the target-domain for the student network. Since SE-GAN combines two promising methods, which are self-ensembling and adversarial training, it gets the advantages from both methods and addresses their respective weaknesses.

All the three discussed domain adaptation methods used SYNTHIA and GTA5 datasets as the labeled source-domain and used Cityscapes dataset as the unlabeled target-domain. The qualitative results on the GTA5→Cityscapes are shown in Figure 6. The quantitative results are presented in Tables 5 and 6. The baseline segmentation models that are directly trained on the source-domain without domain adaptation are presented as "NoAdapt". As noted, the results of domain adaptation methods are better than the results of the

models without domain adaptation. The semantic-edge domain adaptation method has the best MIoU results, which are 55.9% and 52.8% on the SYNTHIA→Cityscapes and the GTA5→Cityscapes, respectively. Generally, the results of all the three methods show great effectiveness of domain adaptation methods in learning domain-invariant representations for semantic segmentation.

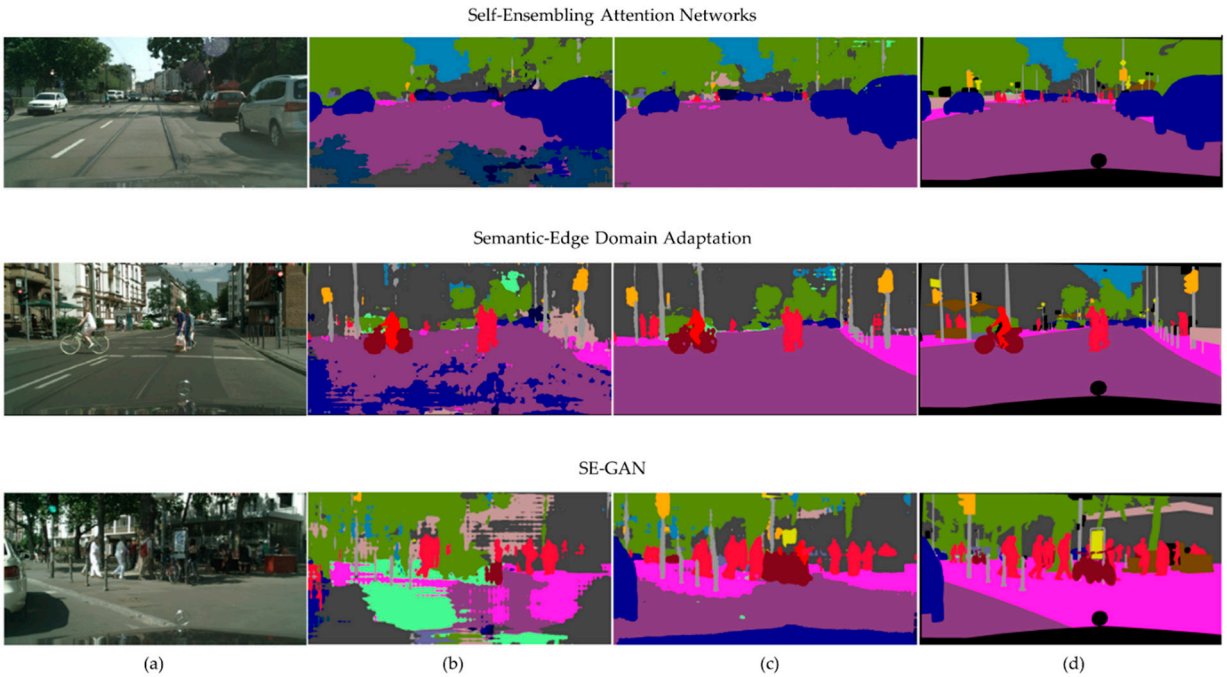


Figure 6. Qualitative results from GTA5 to Cityscapes datasets. (**a**) Input images from Cityscapes. (**b**) Segmentation results without domain adaptation. (**c**) Segmentation results with domain adaptation. (**d**) Ground truth.

Table 5. Evaluation results of semantic segmentation by adapting from SYNTHIA to Cityscapes.

SYNTHIA → Cityscapes			
Method	**Backbone**	**MIoU**	
		NoAdapt	**Adapt**
Self-Ensembling Attention Networks [71]	VGG-16	17.4	37.5
Semantic-Edge Domain Adaptation [72]	ResNet-101	37.8	**55.9**
SE-GAN [73]	ResNet-101	33.3	48.9

Table 6. Evaluation results of semantic segmentation by adapting from GTA5 to Cityscapes.

GTA5 → Cityscapes			
Method	**Backbone**	**MIoU**	
		NoAdapt	**Adapt**
Self-Ensembling Attention Networks [71]	VGG-16	21.2	35.7
Semantic-Edge Domain Adaptation [72]	ResNet-101	36.5	**52.8**
SE-GAN [73]	ResNet-101	37.2	50.1

5. Frameworks

Recently, researchers have been motivated to explore the capabilities of deep network architectures by proposing new frameworks for semantic segmentation. Some frameworks are based on FCNs and others on CNNs. This section overviews some state-of-the-art frameworks for the segmentation task. Table 7 provides some useful information about the reviewed frameworks, such as the framework tasks and evaluation results.

5.1. ParseNet

ParseNet [74] is a deep convolutional network. Figure 7 shows the contexture module overview. The basic idea of this framework is to use the average feature for a layer to augment the features at each location. In addition, multiple studies such as FCN [21] have increased the performance of baseline networks. Moreover, the global feature, along with a technique for learning normalization parameters, has also been proposed by Liu et al. [74] to increase the accuracy, to clarify the local confusion, and to smooth segmentation even over their improved versions of the baselines. With small additional computational cost over baselines, ParseNet achieved state-of-the-art performance on PASCAL-Context [75] and SiftFlow [76] datasets.

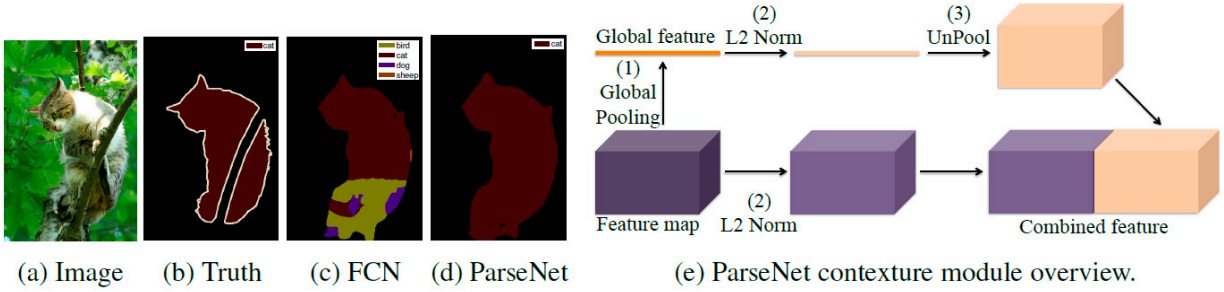


Figure 7. ParseNet framework [74].

5.2. Pyramid Scene Parsing Network (PSPNet)

PSPNet [77] is a framework for pixel-level prediction tasks, as shown in Figure 8. The capability of context information by different-region-based aggression has been exploited through the pyramid pooling module along with the proposed PSPNet. State-of-the-art performance has been achieved by PSPNet on different datasets. It was the first in Cityscapes benchmark, PASCAL VOC 2012 benchmark, and ImageNet scene parsing challenge 2016. Moreover, PSPNet yielded a record of 85.4% Mean Intersection over Union (MIoU) accuracy on PASCAL VOC 2012 and 80.2% accuracy on Cityscapes.

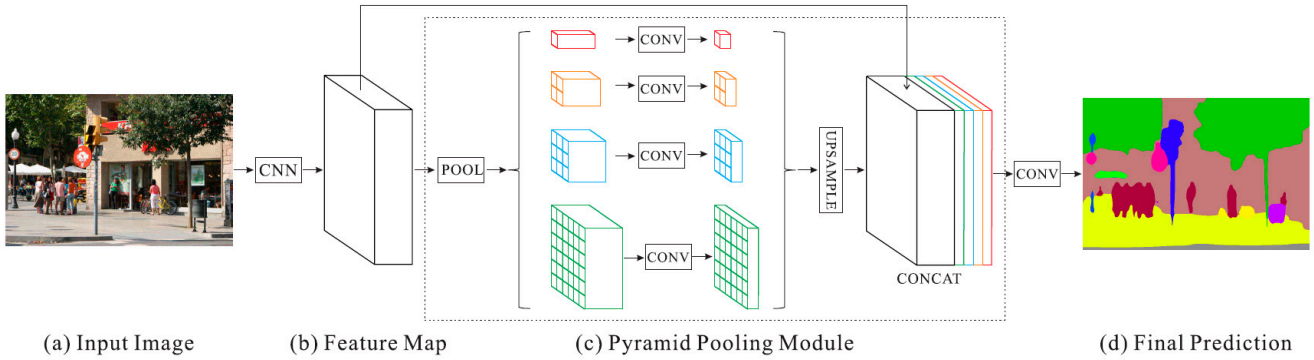


Figure 8. PSPNet framework [77].

Table 7. Information about the reviewed semantic segmentation frameworks in chronological order.

Framework	Tasks	Evaluation Results
ParseNet [74]	Semantic Segmentation	Ranked #46 on semantic segmentation on PASCAL VOC 2012 test
PSPNet [77]	Semantic Segmentation Real-Time Semantic Segmentation Video Semantic Segmentation Lesion Segmentation Scene Parsing Image Classification	Ranked #3 on video semantic segmentation on Cityscapes val
BiSeNet [78]	Semantic Segmentation Real-Time Semantic Segmentation	Ranked #4 on semantic segmentation on SkyScapes-Dense
DeepLabv3+ [79]	Semantic Segmentation Lesion Segmentation Image Classification	Ranked #1 on lesion segmentation on ATLAS
Mask R-CNN [80]	Semantic Segmentation Instance Segmentation 3D Instance Segmentation Human Part Segmentation Nuclear Segmentation Panoptic Segmentation Object Detection Real-Time Object Detection Key Point Detection Multi-Human Parsing Pose Estimation	Ranked #1 on real-time object detection on COCO minival (MAP metric)
DANet [81]	Semantic Segmentation Scene Segmentation	Ranked #8 on semantic segmentation on COCO-Stuff test
HTC [82]	Semantic Segmentation Instance Segmentation Object Detection	Ranked #27 on instance segmentation on COCO test-dev
FastFCN [83]	Semantic Segmentation	Ranked #29 on semantic segmentation on PASCAL Context
GSCNN [84]	Semantic Segmentation	Ranked #16 on semantic segmentation on Cityscapes test
ShelfNet [85]	Semantic Segmentation Real-Time Semantic Segmentation Scene Understanding Autonomous Driving	Ranked #11 on real-time semantic segmentation on Cityscapes test
3D-MiniNet [86]	Semantic Segmentation Real-Time Semantic Segmentation 3D Semantic Segmentation Real-Time 3D Semantic Segmentation LiDAR Semantic Segmentation Autonomous Driving Autonomous Vehicles	Ranked #1 on real-time 3D semantic segmentation on SemanticKITTI
BlendMask [87]	Semantic Segmentation Instance Segmentation Real-Time Instance Segmentation	Ranked #6 on real-time instance segmentation on COCO
HRNet [88]	Semantic Segmentation Instance Segmentation Object Detection Pose Estimation Representation Learning	Ranked #1 on object detection COCO test-dev (Hardware Burden metric)
SANet [89]	Semantic Segmentation	Ranked #14 on semantic segmentation on PASCAL VOC 2012 test (using extra training data)

5.3. Bilateral Segmentation Network (BiSeNet)

BiSeNet [78] is a novel framework. Figure 9 shows its architecture. This framework addresses the problem of compromising spatial resolution to achieve real-time inference speed, which leads to poor performance. Moreover, Yu et al. [78] designed a spatial path with a small stride to preserve the spatial information and to generate high-resolution features. Additionally, a context path with a fast down-sampling strategy is employed to obtain sufficient receptive field. Furthermore, Yu et al. [78] introduced a new feature fusion module, which combines features efficiently on top of the spatial and context paths. However, the BiSeNet framework balances between the segmentation performance and speed on CamVid, Cityscapes, and COCO-Stuff [90] datasets. Particularly, on a Cityscapes test set, the proposed framework [78] achieved 68.4% MIoU accuracy with a speed of 105 frame per second (fps) on one NVIDIA Titan XP card for a 2047 × 1024 input.

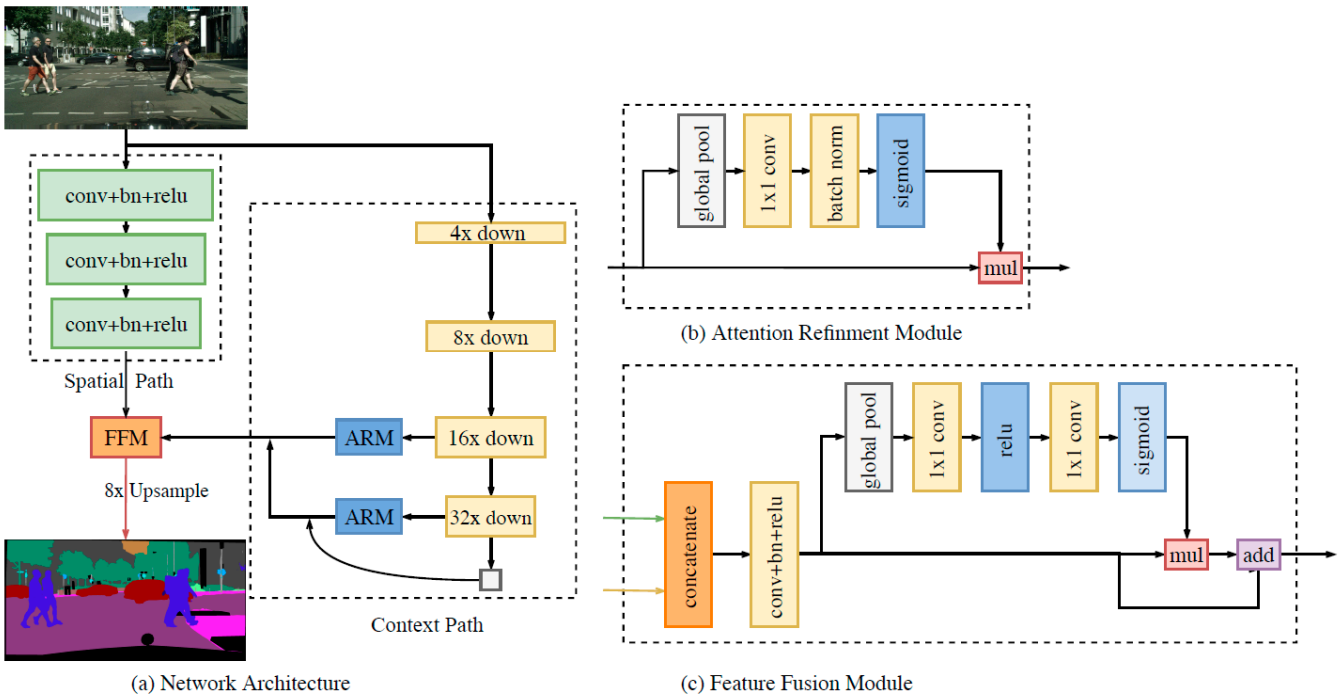


Figure 9. BiSeNet framework [78].

5.4. DeepLabv3+

DeepLabv3+ [79] model combines two methods: the former networks and the latter networks. The former networks encode multiscale contextual information by probing the incoming features with filters or by pooling operations at multiple-effective fields-of-view and multiple rates. The latter networks capture sharper object boundaries by gradually recovering the spatial information. However, the proposed DeepLabv3+ [79] extends DeepLabv3, where an effective decoder module has been added to refine the segmentation results specifically along object boundaries, as shown in Figure 10. Chen et al. [79] explored the Xception model and applied depthwise separable convolution to decoder modules and atrous spatial pyramid pooling, where the result was a stronger and faster encoder-decoder network. On PASCAL VOC 2012 and Cityscapes datasets, DeepLabv3+ achieved 89.0% test set performance and 82.1% without postprocessing, respectively.

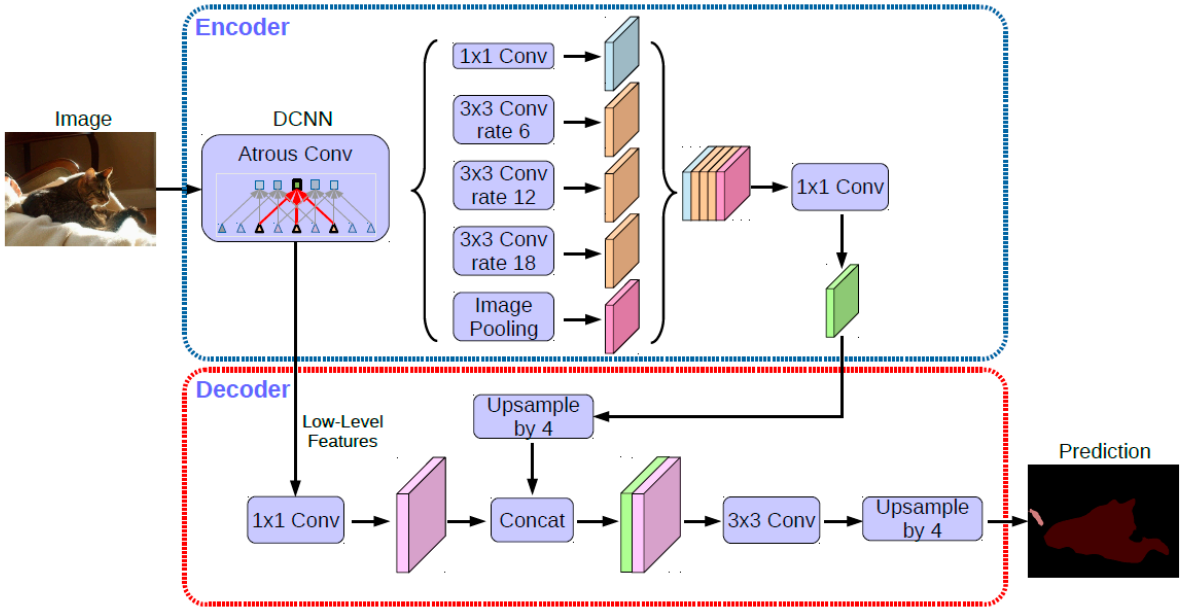


Figure 10. DeepLabv3+ framework [79].

5.5. Mask R-CNN

Mask R-CNN [80] is an approach for object instance segmentation and is shown in Figure 11. Mask R-CNN extends Faster R-CNN [91], where a branch that predicts an object mask was added in parallel with the existing branch for bounding box recognition. The Mask R-CNN framework detects objects in an image and simultaneously generates a high-quality segmentation for each instance. Furthermore, this framework is easy to train and can be generalized to other tasks. The results showed outstanding performance in the COCO challenges in all three tracks: bounding-box object detection, instance segmentation, and person key-point detection. Including the winners of the COCO 2016 challenge, Mask R-CNN surpasses all existing single-model entries on all tasks.

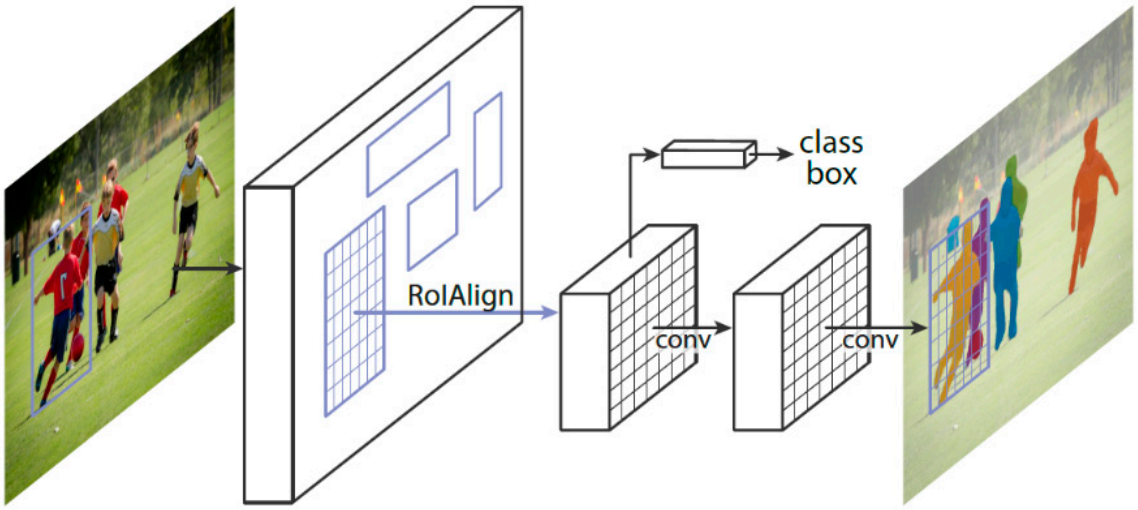


Figure 11. Mask R-CNN framework for instance segmentation [80].

5.6. Dual Attention Network (DANet)

DANet [81] is unlike other frameworks that capture contexts by multiscale feature fusion. Fu et al. [81] captured rich contextual dependencies to address the segmentation task based on the self-attention mechanism. DANet integrates local features with their global

dependencies. Figure 12 shows the framework overview, where two types of attention modules have been appended on top of a dilated FCN. These two modules model the segmentation interdependencies in spatial and channel dimensions. Moreover, at each position, the position attention module selectively aggregates the feature by a weighted sum of all of the features at all positions. Similar features are related to each other, regardless of their distances. On the other hand, the channel attention module integrates associated features among all channel maps in order to selectively emphasize interdependent channel maps. Furthermore, the outputs of the two attention modules are summed for improving feature representation to produce augmentation results that are more accurate. State-of-the-art performance has been achieved by DANet on PASCAL-Context, Cityscapes, and COCO-Stuff datasets. On a Cityscapes test set, DANet achieved 81.5% MIoU without using coarse data.

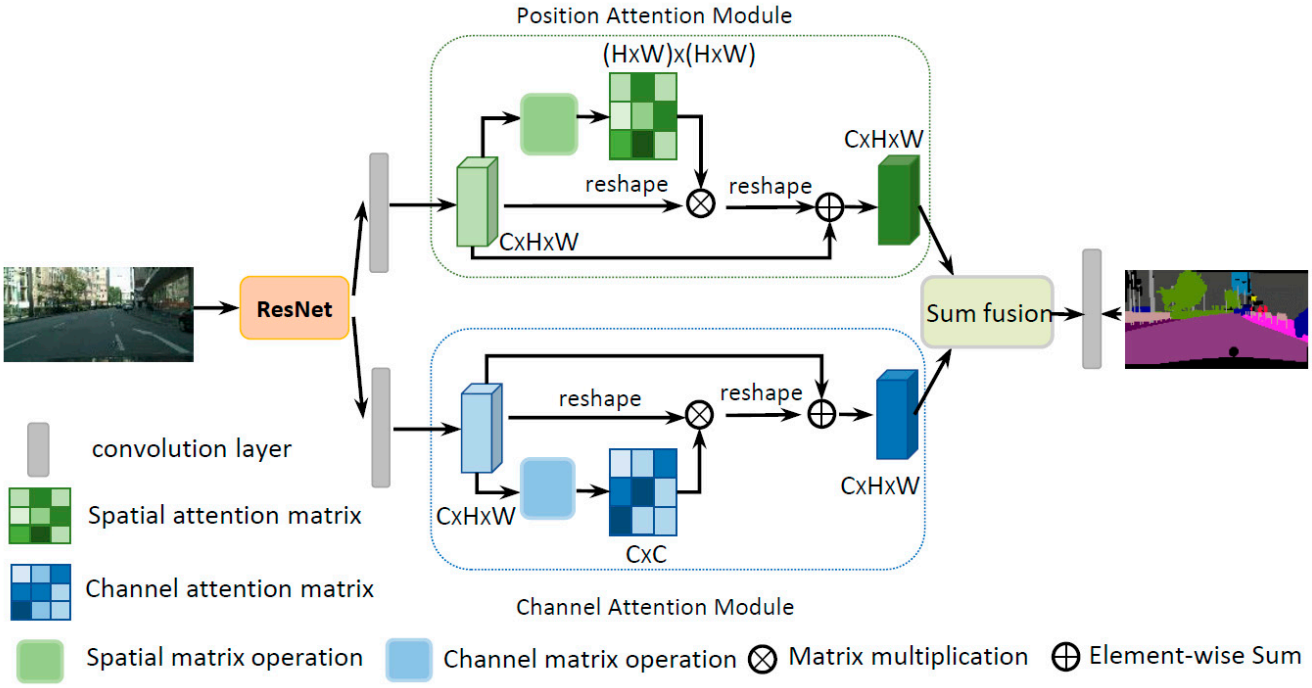


Figure 12. DANet framework [81].

5.7. Hybrid Task Cascade (HTC)

HTC [82] is a new cascade framework for instance segmentation. Chen et al. [82] found that fully leveraging the complementary relationship between detection and segmentation can lead to a successful instance segmentation cascade. Comparing this framework to other existing frameworks, it has differences in multiple aspects. First, instead of executing bounding box regression and mask prediction in parallel, they are interleaved. Second, a direct path is incorporated to reinforce the information flow between mask branches by feeding the mask feature of the previous stage to the current stage. Third, an additional semantic segmentation branch is added and fused with box and mask branches to explore information that is more contextual. As a result of these changes to the architecture of the framework, the information flow improved across stages and between tasks. Overall, the proposed HTC framework achieved a 1.5% improvement on the COCO dataset over a strong Cascade Mask R-CNN baseline.

5.8. FastFCN

FastFCN [83] is a Joint Pyramid Upsampling (JPU) module. It replaced the dilated convolutions that consume time and memory. Moreover, as a joint upsampling problem, the module formulates a function for extracting high-resolution maps. Figure 13 shows the framework overview. Experimental results showed that JPU is considered superior to

other upsampling modules and it can be plugged into existing techniques to improve the performance and to reduce the computation complexity. State-of-the-art performance has been achieved in the PASCAL-Context and ADE20K datasets.

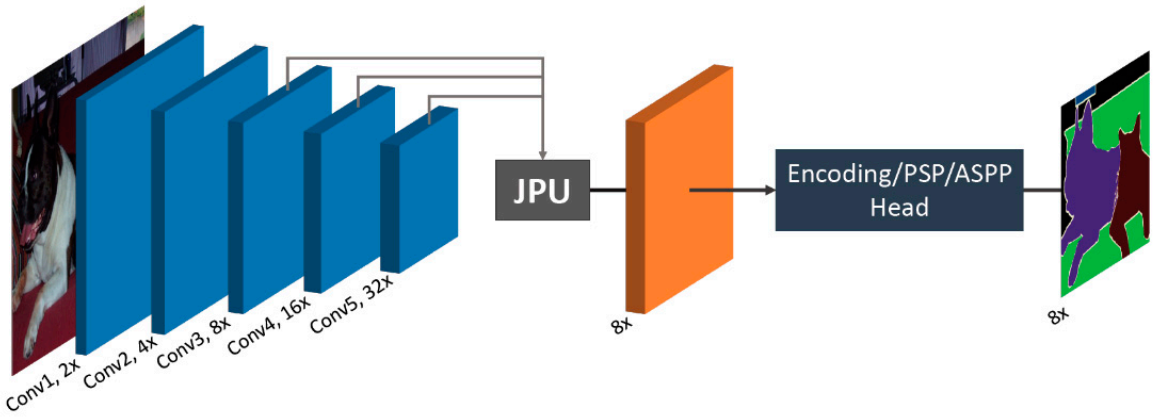


Figure 13. FastFCN framework [83].

5.9. Gated Shape CNN (GSCNN)

GSCNN [84] is a two-stream CNN framework for semantic segmentation; it is shown in Figure 14. GSCNN wires shape information as a separate processing stream, where the shape stream processes information in parallel to the classical stream. This is unlike most state-of-the-art image segmentation methods, which form a dense image representation in which the shape, color, and texture information are processed together inside a deep CNN. Moreover, Takikawa et al. [84] proposed a new gating mechanism that connects all of the intermediate layers of two streams. Specifically, in the classical stream, the higher-level activations are used for gating the lower-level activations in the shape stream. As a result, the noise is removed, and the shape stream is focused only on processing the related boundary information. The Takikawa et al. [84] experiments also showed that with the gating mechanism, it is possible to use a very shallow architecture for the shape stream, which operates on the image-level resolution. This leads to a very effective architecture that generates sharper predictions around object boundaries and remarkably increases performance on smaller and thinner objects. GSCNN framework achieved state-of-the-art performance on Cityscapes dataset, improving accuracy and boundary quality over strong baselines by 2% MIoU and 4% F-score, respectively.

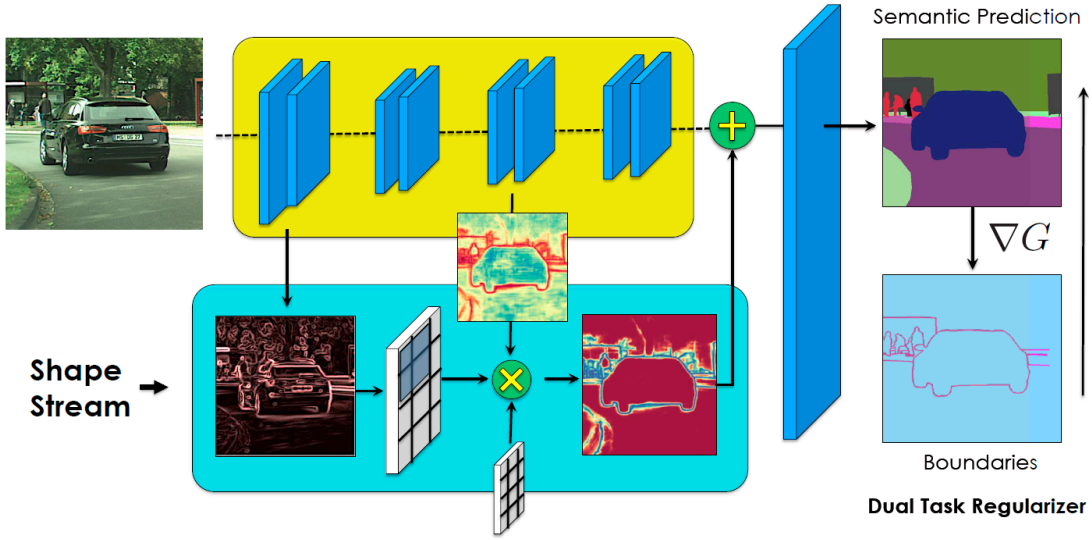


Figure 14. GSCNN framework [84].

5.10. ShelfNet

ShelfNet [85] is a novel, shelf-shaped framework for accurate fast-semantic segmentation. Figure 15 shows the structure of ShelfNet, which has various encoder–decoder branch pairs with skip connections at each spatial level that look like a shelf with different columns. The unique shelf-shaped structure can be viewed as a group of different deep and shallow paths. Zhuang et al. [85] reduced the channel number to reduce the computation complexity and achieved high accuracy with their unique structure. Moreover, a shared-weight strategy has been proposed by Zhuang et al. [85] in the residual block to reduce the parameter number without sacrificing the performance. Compared with PSPNet, ShelfNet achieved inference speed four times faster with similar accuracy on PASCAL VOC dataset. Compared with BiSeNet, which is a real-time segmentation model, at a comparable speed, ShelfNet achieved higher accuracy on Cityscapes dataset. Particularly, ShelfNet with a ResNet-34 backbone achieved 79% MIoU on Cityscapes dataset, surpassing BiSeNet with large ResNet-101 backbone. Experiments by Zhuang et al. [85] proved the outstanding performance of ShelfNet in some applications, such as understanding the street scenes for autonomous driving, which requires speed.

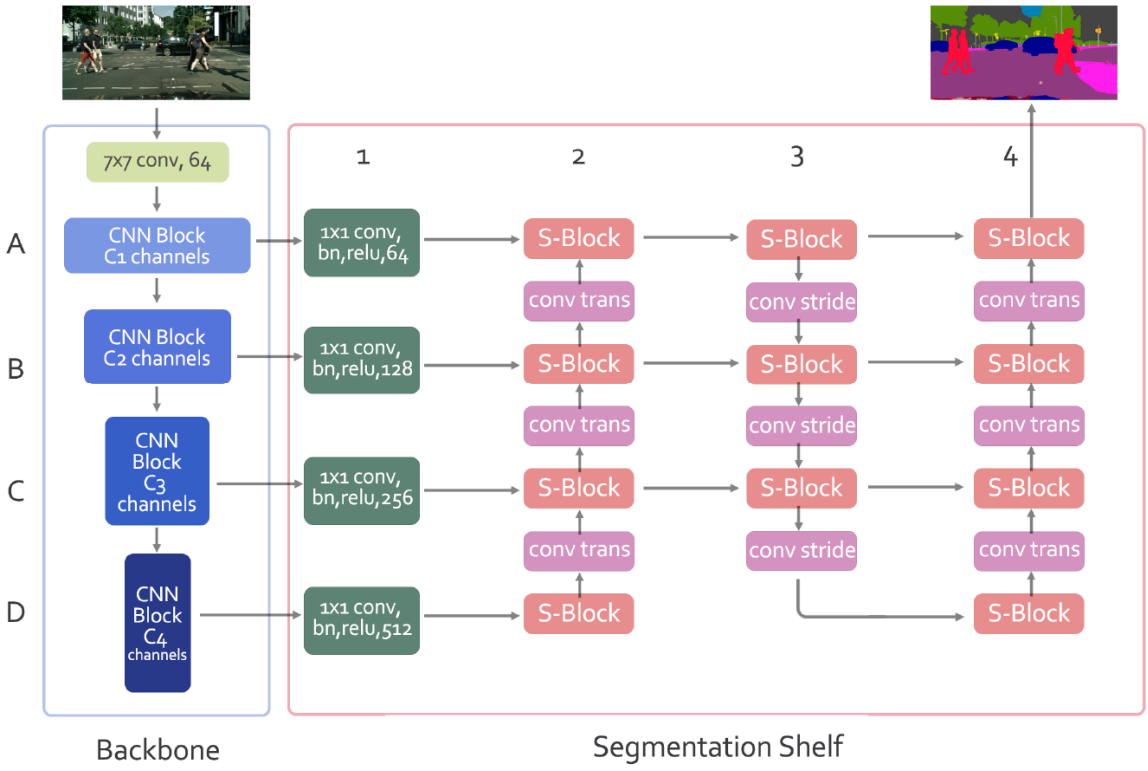


Figure 15. The structure of ShelfNet. Rows A-D represent different spatial levels. Columns 1-4 represent different branches: 3 is encoder branch, 2 and 4 are decoder branches, and 1 reduces the number of channels [85].

5.11. 3D-MiniNet

3D-MiniNet [86] is a novel framework for LiDAR semantic segmentation. 3D-MiniNet is a combination of 2D and 3D learning layers. Figure 16 shows the framework overview. From the 3D data, a novel projection extracts global and local information and then from the raw points, the framework learns a 2D representation. After that, the representation is fed to an efficient 2D Fully Convolutional Neural Network (FCNN) that produces a 2D semantic segmentation. Finally, a postprocessing module enhances and reprojects the produced 2D semantic labels back to the 3D space. The projection learning module is the main novelty in this framework. 3D-MiniNet achieved state-of-the-art results on KITTI and

SemanticKITTI datasets. Compared with the existing methods, 3D-MiniNet is faster and more parameter efficient.

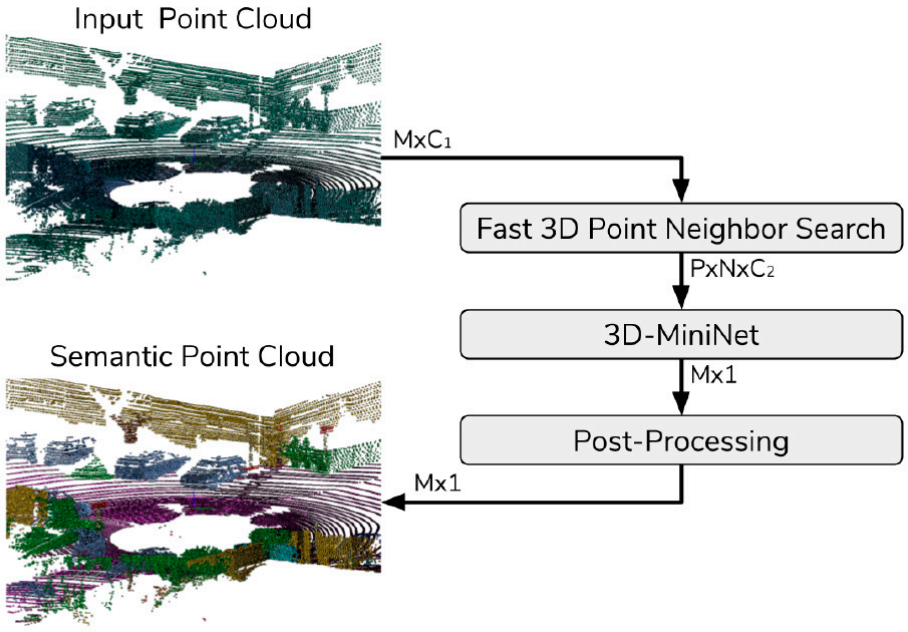


Figure 16. 3D-MiniNet framework [86].

5.12. BlendMask

BlendMask [87] is an improved mask prediction for instance segmentation. It combines instance-level information that has semantic information with lower-level fine-granularity. Figure 17 shows the BlendMask pipeline, which is built upon the state-of-the-art Fully Convolutional One-Stage (FCOS) object detection [92] with minimal changes. Moreover, with few channels, BlendMask predicts dense per-pixel position-sensitive instance features and learns attention maps with simply one convolutional layer for each instance. BlendMask is 20% faster compared to two-stage Mask R-CNN under the same training schedule and can be integrated with state-of-the-art, one-stage detection frameworks. BlendMask is an efficient and simple framework that can be used as a baseline for instance segmentation tasks.

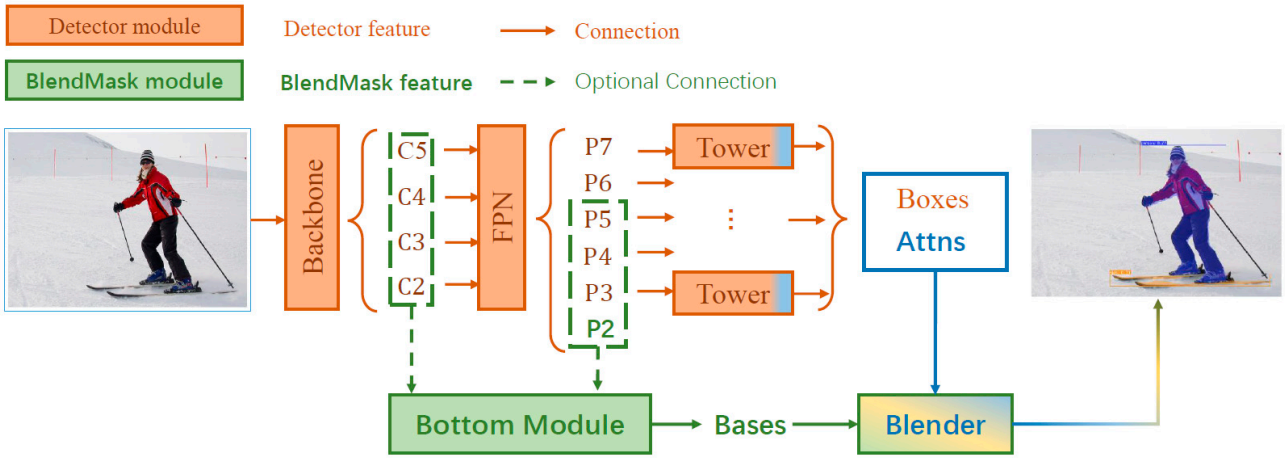


Figure 17. BlendMask pipeline [87].

5.13. High-Resolution Network (HRNet)

HRNet [88] has been proposed for maintaining high-resolution representations for critical computer vision problems such as object detection and semantic segmentation. Most

of the existing state-of-the-art frameworks encode the input images as a low-resolution representation by connecting high-to-low resolution convolutions in series through a subnetwork such as VGG and ResNet. Then, from the encoded low-resolution representation, the high-resolution representation is recovered. HRNet has two main characteristics: (a) the high-to-low resolution convolution streams are connected in parallel, (b) the information across resolution is exchanged frequently. As a result, the representation of HRNet compared to other existing frameworks is spatially more precise and semantically richer. Wang et al. [88] showed the superiority of the proposed HRNet in various applications and suggested that HRNet is a strong backbone for computer vision problems.

5.14. Squeeze-and-Attention Network (SANet)

SANet [89], shown in Figure 18, has been proposed to leverage an effective Squeeze-and-Attention (SA) module. This module accounts for two unique segmentation features: pixel-group attention and pixel-wise prediction. Moreover, the proposed SA module introduced an "attention" convolutional channel for imposing pixel-group attention on convolutional convolution, thus considering spatial-channel interdependencies in an efficient way. Zhong et al. [89] produced the final segmentation results by combining outputs from four hierarchical stages of a SANet for integrating multiscale contexts to obtain an enhanced pixel-wise prediction. Their empirical experiments improved the effectiveness of the SANet framework, where it achieved state-of-the-art performance on PASCAL-Context with 54.4% MIoU and achieved 83.2% MIoU on PASCAL VOC without COCO pretraining.

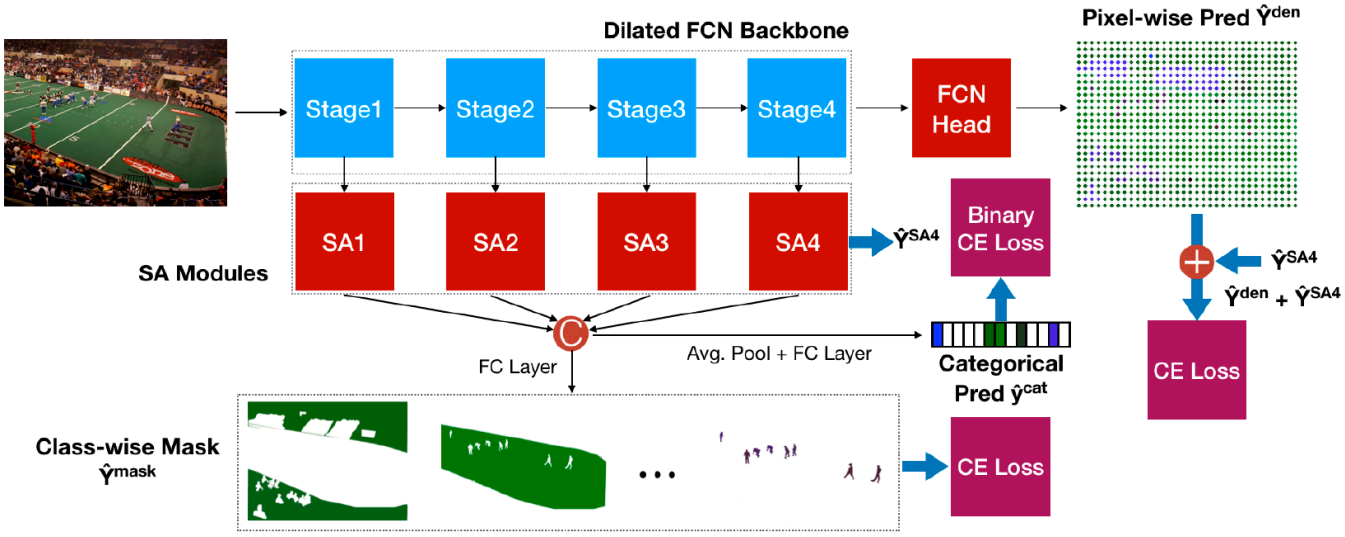


Figure 18. SANet framework [89].

6. Discussion

When reviewing any framework, it is important to take quantitative results into account. This section discusses some of the most popular evaluation metrics used for measuring semantic segmentation systems' performance, in addition to providing the results of the reviewed frameworks on some standard datasets using one of the described evaluation metrics. At last, the results are summarized and concluded.

6.1. Evaluation Metrics

The evaluation of semantic segmentation systems must be performed using popular and standard evaluation metrics which enable fair comparisons with different architectures and frameworks. Moreover, various important aspects, such as execution time, accuracy, and memory footprint, must be evaluated for asserting the usefulness and validity of a system [6]. Based on the context or the purpose of a system, some metrics might be more important than others. For example, in real-world applications, the accuracy can be

expendable up to a certain point in favor of execution time. However, for any proposed architecture or framework, it is important to provide all of the possible metrics.

6.1.1. Execution Time

Most systems should meet specific requirements, especially on the time they spent on inference pass. That is why execution time is a valuable metric. In some situations, knowing the time needed for training a system might be useful, but generally, it is not important. Moreover, knowing the exact timings for any proposed architecture or framework can be meaningless because timings are dependent on the backend implementation and the hardware. Providing timings with a thorough description of the hardware on which the system was executed, along with the benchmark conditions, is very helpful for fellow researchers. In this way, others can estimate if the architecture or the framework is useful for their application. Fair comparisons can be made only under the same conditions for checking which is the fastest.

6.1.2. Accuracy

For assessing the accuracy of any semantic segmentation technique, multiple evaluation criteria have been proposed and used in the literature. For semantic segmentation, currently, there are some well-known metrics for measuring per-pixel labeling techniques. These metrics are discussed in Table 8. For the explanation, the notation details are as follows: assuming a total of $k + 1$ classes from L_0 to L_k, including background or a void class, and the amount of pixels is P_{ij} of class i, which belongs to class j. The number of true positives is represented by P_{ii}, false positives is represented by P_{ij}, and false negatives is represented by P_{ji}, even though the sum of both false positives and false negatives can be either of them [6].

Table 8. The most popular semantic segmentation accuracy metrics.

Metrics	Formula	Evaluation Focus
Pixel Accuracy (PA)	$PA = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}}$	This metric computes the ratio between the amount of correctly classified pixels and the total number of pixels.
Mean Pixel Accuracy (MPA)	$MPA = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}}$	This metric computes the ratio of correct pixels in a per-class basis and the averages over the total number of classes. It is an improved PA.
Mean Intersection over Union (MIoU)	$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}$	This metric computes a ratio between the intersection and the union of two sets, which are the ground truth and the predicted segmentation. Reformulating the ratio is possible as the number of true positives over the sum of true positives, false negatives, and false positives which represents Intersection over Union (IoU). The IoU is computed on a per-class basis and then averaged.
Frequency Weighted Intersection over Union (FWIoU)	$FWIoU = \frac{1}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \sum_{i=0}^{k} \frac{\sum_{j=0}^{k} p_{ij} p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}$	This metric weighs every class importance based on their appearance frequency.

Table 8 discusses four metrics, where the MIoU metric stands out because of its simplicity and representativeness, which make it the most used metric in many competitions and by many researchers for reporting their results.

6.1.3. Memory Footprint

For segmentation techniques, memory usage is an important aspect. Unlike execution time, memory capacity is scalable, and it can be a limiting element. In certain cases,

memory can be insufficient, such as onboard chips for robotic systems compared to a high-performance server. Furthermore, for accelerating deep networks, high-end Graphics Processing Units (GPUs) [93] are used. Considering the same aspects for implementation-dependent as with execution time, and documenting the average and the peak memory footprint of a technique with the description of the complete execution conditions can be extremely helpful for other researchers.

6.2. Results

When reporting the results, remarking the heterogeneity of the papers is very important. Most of the reviewed papers evaluated their frameworks' results on standard datasets and provided enough information to reproduce their results. However, some papers failed to express their results in popular metrics, which made it difficult to make fair comparisons. Moreover, many papers do not provide any information about the execution time or memory footprint. In certain situations, that information is included, but not enough information is provided to reproduce the results, which makes it impossible to know the setups for producing their results, and thus, the information is of no use.

Twelve datasets were selected in this paper: CamVid, KITTI, COCO, PASCAL VOC, Cityscapes, SYNTHIA, GTA5, Mapillary Vistas, ADE20K, SemanticKITTI, nuScenes, and ApolloScape. Some of those datasets cover a wide range of cases and targets, and some are considered new in the field. The accuracy results of the reviewed frameworks are divided into three groups, based on the task of the frameworks: semantic segmentation, 3D semantic segmentation, or real-time semantic segmentation. Table 9 shows semantic segmentation and 3D semantic segmentation results, while Table 10 shows real-time semantic segmentation results. Both tables report the accuracy results of the reviewed frameworks, ordered from the highest to lowest scorer for each dataset. All of the reported results are taken from the frameworks' original papers, where the MIoU metric has been used for evaluation.

Table 9. Semantic segmentation and 3D semantic segmentation accuracy results of the reviewed frameworks ordered from highest to lowest scorer for each dataset.

Dataset	Framework	Backbone	MIoU
Semantic Segmentation			
CamVid [30]	PSPNet		69.1
	BiSeNet	ResNet-18	68.7
PASCAL VOC 2012 [34]	DeepLabv3+	Xception-JFT	89
	SANet *	ResNet-101	86.1
	PSPNet *	ResNet-101	85.4
	ShelfNet *	ResNet-101	84.2
	SANet	ResNet-101	83.2
	DANet	ResNet-101	82.6
	ShelfNet	ResNet-101	81.1
	ParseNet		69.8
Cityscapes test [5]	GSCNN		82.8
	DeepLabv3+ (coarse)		82.1
	DANet	ResNet-101	81.5
	PSPNet (fine and coarse)	ResNet-101	80.2
	ShelfNet	ResNet-34	79
	BiSeNet	ResNet-101	78.9
ADE20K [40]	PSPNet	ResNet-269	44.94
	FastFCN	ResNet-101	44.34
3D Semantic Segmentation			
SemanticKITTI [41]	3D-MiniNet		55.8

* Pretrained on COCO.

Table 10. Real-time semantic segmentation accuracy results of the reviewed frameworks, ordered from highest to lowest scorer for each dataset.

Dataset	2D\3D	Framework	Backbone	Parameters	MIoU	Time	Frame
Cityscapes test [5]	2D	ShelfNet	ResNet-18	23.5 M	74.8	16.9 ms	59.2 fps
	2D	BiSeNet	ResNet-18	49.0 M	74.7	15.2 ms	65.5 fps
SemanticKITTI [41]	3D	3D-MiniNet		3.97 M	55.8		28 fps

6.2.1. Semantic Segmentation

The first dataset is CamVid, which is considered one of the most important urban scenes datasets. The reviewed frameworks that provided accuracy metrics for the CamVid dataset were PSPNet and BiSeNet. The top scorer was PSPNet with a 69.1% MIoU. PSPNet is an effective network for complex scene understanding.

The second dataset is PASCAL VOC 2012, a general-purpose dataset that has been used to evaluate many deep learning techniques. The results showed a great improvement from the ParseNet framework to the top scorer, which was DeepLabv3+ (Xception-JFT) with an 89% MIoU. DeepLabv3+ achieved state-of-the-art performance on PASCAL VOC 2012.

The third dataset is Cityscapes, which is one of the most in use and challenging datasets. The top framework on this dataset was GSCNN with an 82.8% MIoU. GSCNN was not trained on coarse data and outperformed very strong frameworks trained on extra coarse data such as DeepLabv3+ and PSPNet.

Finally, the results of the top scorer on ADE20K, which is one of the largest available datasets, was PSPNet with a 44.94% MIoU. The deep network of ResNet-269 used as a backbone helps to increase the performance of PSPNet.

6.2.2. 3D Semantic Segmentation

The only reviewed framework used for the 3D semantic segmentation task was 3D-MiniNet. It achieved 55.8% MIoU on the SemanticKITTI dataset. 3D-MiniNet is efficient and fast for 3D LIDAR semantic segmentation.

6.2.3. Real-Time Semantic Segmentation

For Cityscapes dataset, ShelfNet and BiSeNet frameworks were used for the real-time semantic segmentation task. Both frameworks scored similar results, 74.8% and 74.7% MIoU, respectively. ShelfNet achieved a comparable inference speed as when compared to BiSeNet.

Similar to 3D semantic segmentation, the only reviewed framework used for the 3D real-time semantic segmentation task was 3D-MiniNet. It achieved 55.8% MIoU on the SemanticKITTI dataset and showed state-of-the-art performance using few parameters (3.97 M).

6.3. Summary

In considering the results, the most important conclusion is relevant to reproducibility. As reviewed in this paper, some frameworks have not been tested on standard datasets. For that reason, it is impossible to make fair comparisons. Moreover, some papers do not include descriptions for their experiments' setups, which significantly hurt the reproducibility. All proposed frameworks must use standard datasets for reporting their results and publicly share exhaustive descriptions and weights to enable reproducibility and further progress.

Furthermore, regarding metrics such as memory footprint and execution time, there is a lack of information about them in almost all reviewed papers. The reason for that is most papers focused on the accuracy of their proposed frameworks and did not consider time or memory. It is important to know where those frameworks are applied. Most of them ran on embedded devices, such as robots [94,95], drones, and autonomous cars, which usually have limited memory and computational power.

Based on the results, PSPNet was the most tested framework. It has been tested on four standard datasets and outperformed other frameworks on two datasets. In addition, DeepLabv3+ is a solid framework that scored high results on the two datasets it has been tested on. Regarding the new frameworks that have been proposed recently, such as SANet, ShelfNet, and GSCNN, they all achieved great accuracy results and significantly improved over strong baselines. However, for the 3D semantic segmentation and real-time semantic segmentation, a lot of work is needed for future research on preprocessing, dealing with 3D data, and remaking frameworks with higher power and flexibility.

7. Future Directions

Depending on the reviewed papers that present state-of-the-art on the field, some future research directions are highlighted in this section.

- Memory: For segmentation networks, significant amounts of memory are needed for execution. Some devices have limited memory, where the networks must be simplified to fit in them. Network simplification is made by reducing complexity, which decreases accuracy. One of the most promising research directions is simplifying a network by reducing its weight and keeping the accuracy of the original network architecture [96–98].
- 3D Datasets: There is an urgent need for large-scale 3D datasets, due to the evaluation of new segmentation techniques that depend on 3D data. Even though there are some promising works, a need remains for varied and better data. It is important to create a 3D dataset from real data because most of the existing 3D datasets are synthetic [99–101].
- Real-Time Segmentation: Most segmentation implementations are far from the common camera framerate, which is at least 25 fps. Currently, many of the existing frameworks take between 100 ms to 500 ms to process low-resolution images. For that reason, there is a need for new works that focus on real-time segmentation while finding a trade-off between runtime and accuracy [102–104].

8. Conclusions

This paper focused on semantic segmentation using deep learning, particularly for autonomous driving. It covered well-known and some recent work on the field in addition to the basic background knowledge about deep learning for semantic segmentation tasks. It compared 14 frameworks, 12 datasets, and different data augmentation and domain adaptation techniques. Furthermore, frameworks were reviewed, and their tasks were stated. Datasets, along with their characteristics and purposes, were described. The benefits of data augmentation and domain adaptation techniques and their experimental results were discussed. Additionally, the results of the frameworks and datasets were presented in tabular form. Finally, a discussion about the results was provided and future research directions in the field were suggested. In conclusion, there are many successful implementations for semantic segmentation, but it is still a problem that needs additional improved solutions that can be applied to real-world critical applications. Moreover, deep learning is a powerful tool to tackle semantic segmentation problems. Many techniques and innovations for autonomous driving systems are expected to be proposed in the coming years.

Author Contributions: Conceptualization, M.B.A. and H.A.; resources, H.A.; writing—original draft preparation, H.A.; writing—review and editing, M.B.A. and H.A.; supervision, M.B.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Project No. Grant912].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Oberweger, M.; Wohlhart, P.; Lepetit, V. Hands Deep in Deep Learning for Hand Pose Estimation. *arXiv* **2015**, arXiv:1502.06807.
2. Wan, J.; Wang, D.; Hoi, S.C.H.; Wu, P.; Zhu, J.; Zhang, Y.; Li, J. Deep Learning for Content-Based Image Retrieval: A Comprehensive Study. In Proceedings of the 22nd ACM International Conference on Multimedia, New York, NY, USA, 3–7 November 2014; pp. 157–166.
3. Ess, A.; Müller, T.; Ch, M.; Grabner, H.; van Gool, L.; Leuven Belgium, K.U. Segmentation-Based Urban Traffic Scene Understanding. In Proceedings of the 2009 British Machine Vision Conference, London, UK, 7–10 September 2009; p. 2.
4. Geiger, A.; Lenz, P.; Urtasun, R. Are We Ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
5. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 3213–3223.
6. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv* **2017**, arXiv:1704.06857.
7. Kirillov, A.; He, K.; Girshick, R.; Rother, C.; Dollar, P. Panoptic Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9396–9405.
8. Cheng, B.; Collins, M.D.; Zhu, Y.; Liu, T.; Huang, T.S.; Adam, H.; Chen, L.C. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12475–12485.
9. Broggi, A.; Berte, S. Vision-Based Road Detection in Automotive Systems: A Real-Time Expectation-Driven Approach. *J. Artif. Intell. Res.* **1995**, *3*, 325–348. [CrossRef]
10. Jyothi, S.; Padmavati, S.; Visvavidyalayam, M. A Survey on Threshold Based Segmentation Technique in Image Processing. *Int. J. Innov. Res.* **2014**, *3*, 234–239.
11. Nath, S.S.; Mishra, G.; Kar, J.; Chakraborty, S.; Dey, N. A Survey of Image Classification Methods and Techniques. In Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, Kanyakumari District, India, 10–11 July 2014; pp. 554–557.
12. Gulhane, A.; Paikrao, P.L.; Chaudhari, D.S. A Review of Image Data Clustering Techniques. *Int. J. Soft Comput. Eng.* **2012**, *2*, 212–215.
13. Olaode, A.; Naghdy, G.; Todd, C. Unsupervised Classification of Images: A Review. *Int. J. Image Process.* **2014**, *8*, 325–342.
14. Peng, B.; Zhang, L.; Zhang, D. A survey of graph theoretical approaches to image segmentation. *Pattern Recognit.* **2013**, *46*, 1020–1038. [CrossRef]
15. Prieto, A.; Prieto, B.; Ortigosa, E.M.; Ros, E.; Pelayo, F.; Ortega, J.; Rojas, I. Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing* **2016**, *214*, 242–268. [CrossRef]
16. Ning, F.; Delhomme, D.; LeCun, Y.; Piano, F.; Bottou, L.; Barbano, P.E. Toward automatic phenotyping of developing embryos from videos. *IEEE Trans. Image Process.* **2005**, *14*, 1360–1371. [CrossRef]
17. Ciresan, D.; Giusti, A.; Gambardella, L.; Schmidhuber, J. Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 2843–2851.
18. Farabet, C.; Couprie, C.; Najman, L.; LeCun, Y. Learning Hierarchical Features for Scene Labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 1915–1929. [CrossRef] [PubMed]
19. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous Detection and Segmentation. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 297–312.
20. Gupta, S.; Girshick, R.; Arbeláez, P.; Malik, J. Learning Rich Features from RGB-D Images for Object Detection and Segmentation. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 345–360.
21. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
22. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
23. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
24. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
25. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
27. Jégou, S.; Drozdzal, M.; Vazquez, D.; Romero, A.; Bengio, Y. 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, Honolulu, HI, USA, 21–26 July 2017; pp. 11–19.

28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
29. Choi, S.; Kim, J.T.; Choo, J. Cars Can't Fly up in the Sky: Improving Urban-Scene Segmentation via Height-Driven Attention Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9373–9383.
30. Brostow, G.J.; Fauqueur, J.; Cipolla, R. Semantic Object Classes in Video: A High-Definition Ground Truth Database. *Pattern Recognit. Lett.* **2009**, *30*, 88–97. [CrossRef]
31. Sturgess, P.; Alahari, K.; Ladicky, L.; Torr, P.H.S. Combining Appearance and Structure from Motion Features for Road Scene Understanding. In Proceedings of the 2009 British Machine Vision Conference, London, UK, 7–10 September 2009.
32. Alvarez, J.M.; Gevers, T.; Lecun, Y.; Lopez, A.M. Road Scene Segmentation from a Single Image. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; pp. 376–389.
33. Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
34. Everingham, M.; Eslami, S.M.A.; van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
35. Ros, G.; Ramos, S.; Granados, M.; Bakhtiary, A.; Vazquez, D.; Lopez, A.M. Vision-Based Offline-Online Perception Paradigm for Autonomous Driving. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 5–9 January 2015; pp. 231–238.
36. Zhang, R.; Candra, S.A.; Vetter, K.; Zakhor, A. Sensor Fusion for Semantic Segmentation of Urban Scenes. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 1850–1857.
37. Ros, G.; Sellart, L.; Materzynska, J.; Vazquez, D.; Lopez, A.M. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3234–3243.
38. Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. Playing for Data: Ground Truth from Computer Games. In Proceedings of the 2016 European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 102–118.
39. Neuhold, G.; Ollmann, T.; Rotabuì, S.; Kontschieder, P. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 4990–4999.
40. Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; Torralba, A. Scene Parsing through ADE20K Dataset. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 633–641.
41. Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 9297–9307.
42. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. NuScenes: A Multimodal Dataset for Autonomous Driving. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
43. Huang, X.; Wang, P.; Cheng, X.; Zhou, D.; Geng, Q.; Yang, R. The ApolloScape Open Dataset for Autonomous Driving and Its Application. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2702–2719. [CrossRef]
44. Brostow, G.J.; Shotton, J.; Fauqueur, J.; Cipolla, R. Segmentation and Recognition Using Structure from Motion Point Clouds. In Proceedings of the 10th European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 44–57.
45. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision Meets Robotics: The KITTI Dataset. *Int. J. Rob. Res.* **2013**, *32*, 1231–1237. [CrossRef]
46. Ros, G.; Alvarez, J.M. Unsupervised Image Transformation for Outdoor Semantic Labelling. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium, Seoul, Korea, 28 June–1 July 2015; pp. 537–542.
47. Wong, S.C.; Gatt, A.; Stamatescu, V.; McDonnell, M.D. Understanding Data Augmentation for Classification: When to Warp? In Proceedings of the 2016 International Conference on Digital Image Computing: Techniques and Applications, Gold Coast, Australia, 30 November–2 December 2016; pp. 1–6.
48. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. In Proceedings of the 28th Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
49. Bagherinezhad, H.; Horton, M.; Rastegari, M.; Farhadi, A. Label Refinery: Improving ImageNet Classification through Label Progression. *arXiv* **2018**, arXiv:1805.02641.
50. Taylor, L.; Nitschke, G. Improving Deep Learning Using Generic Data Augmentation. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, Bangalore, India, 18–21 November 2018; pp. 1542–1547.
51. Fei-Fei, L.; Fergus, R.; Perona, P. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop, Washington, DC, USA, 27 June–2 July 2004; p. 178.
52. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random Erasing Data Augmentation. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 13001–13008.

53. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
54. Kang, G.; Dong, X.; Zheng, L.; Yang, Y. PatchShuffle Regularization. *arXiv* **2017**, arXiv:1707.07103.
55. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. Master's Thesis, Computer Science Department, University of Toronto, Toronto, ON, Canada, 2009.
56. DeVries, T.; Taylor, G.W. Dataset Augmentation in Feature Space. *arXiv* **2017**, arXiv:1702.05538.
57. LeCun, Y. The MNIST Database of Handwritten Digits. Available online: http://yann.lecun.com/exdb/mnist/ (accessed on 8 April 2022).
58. Gatys, L.A.; Ecker, A.S.; Bethge, M. A Neural Algorithm of Artistic Style. *arXiv* **2015**, arXiv:1508.06576. [CrossRef]
59. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In Proceedings of the 14th European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 694–711.
60. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
61. dos Tanaka, F.H.K.S.; Aranha, C. Data Augmentation Using GANs. *arXiv* **2019**, arXiv:1904.09135.
62. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2642–2651.
63. Antoniou, A.; Storkey, A.; Edwards, H. Data Augmentation Generative Adversarial Networks. *arXiv* **2017**, arXiv:1711.04340.
64. Mariani, G.; Scheidegger, F.; Istrate, R.; Bekas, C.; Malossi, C. BAGAN: Data Augmentation with Balancing GAN. *arXiv* **2018**, arXiv:1803.09655.
65. Yi, X.; Walia, E.; Babyn, P. Generative Adversarial Network in Medical Imaging: A Review. *Med. Image Anal.* **2019**, *58*, 101552. [CrossRef]
66. Shijie, J.; Ping, W.; Peiyi, J.; Siping, H. Research on Data Augmentation for Image Classification Based on Convolution Neural Networks. In Proceedings of the 2017 Chinese Automation Congress, Jinan, China, 20–22 October 2017; pp. 4165–4170.
67. Tzeng, E.; Hoffman, J.; Zhang, N.; Saenko, K.; Darrell, T. Deep Domain Confusion: Maximizing for Domain Invariance. *arXiv* **2014**, arXiv:1412.3474.
68. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial Discriminative Domain Adaptation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 7167–7176.
69. Long, M.; Cao, Y.; Wang, J.; Jordan, M.I. Learning Transferable Features with Deep Adaptation Networks. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 97–105.
70. Hoffman, J.; Wang, D.; Yu, F.; Darrell, T. FCNs in the Wild: Pixel-Level Adversarial and Constraint-Based Adaptation. *arXiv* **2016**, arXiv:1612.02649.
71. Xu, Y.; Du, B.; Zhang, L.; Zhang, Q.; Wang, G.; Zhang, L. Self-Ensembling Attention Networks: Addressing Domain Shift for Semantic Segmentation. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 5581–5588.
72. Chen, H.; Wu, C.; Xu, Y.; Du, B. Unsupervised Domain Adaptation for Semantic Segmentation via Low-Level Edge Information Transfer. *arXiv* **2021**, arXiv:2109.08912.
73. Xu, Y.; He, F.; Du, B.; Zhang, L.; Tao, D. Self-Ensembling GAN for Cross-Domain Semantic Segmentation. *arXiv* **2021**, arXiv:2112.07999.
74. Liu, W.; Rabinovich, A.; Berg, A.C. ParseNet: Looking Wider to See Better. *arXiv* **2015**, arXiv:1506.04579.
75. Mottaghi, R.; Chen, X.; Liu, X.; Cho, N.-G.; Lee, S.-W.; Fidler, S.; Urtasun, R.; Yuille, A. The Role of Context for Object Detection and Semantic Segmentation in the Wild. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 891–898.
76. Liu, C.; Yuen, J.; Torralba, A. SIFT Flow: Dense Correspondence across Scenes and Its Applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 978–994. [CrossRef] [PubMed]
77. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 2881–2890.
78. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 325–341.
79. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the 15th European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 801–818.
80. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
81. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
82. Chen, K.; Pang, J.; Wang, J.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Shi, J.; Ouyang, W.; et al. Hybrid Task Cascade for Instance Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4974–4983.

83. Wu, H.; Zhang, J.; Huang, K.; Liang, K.; Yu, Y. FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation. *arXiv* **2019**, arXiv:1903.11816.
84. Takikawa, T.; Acuna, D.; Jampani, V.; Fidler, S. Gated-SCNN: Gated Shape CNNs for Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 5229–5238.
85. Zhuang, J.; Yang, J.; Gu, L.; Dvornek, N. ShelfNet for Fast Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019; pp. 847–856.
86. Alonso, I.; Riazuelo, L.; Montesano, L.; Murillo, A.C. 3D-MiniNet: Learning a 2D Representation from Point Clouds for Fast and Efficient 3D LIDAR Semantic Segmentation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5432–5439. [CrossRef]
87. Chen, H.; Sun, K.; Tian, Z.; Shen, C.; Huang, Y.; Yan, Y. BlendMask: Top-Down Meets Bottom-Up for Instance Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8573–8581.
88. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef]
89. Zhong, Z.; Lin, Z.Q.; Bidart, R.; Hu, X.; Daya, I.B.; Li, Z.; Zheng, W.-S.; Li, J.; Wong, A. Squeeze-and-Attention Networks for Semantic Segmentation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13065–13074.
90. Caesar, H.; Uijlings, J.; Ferrari, V. COCO-Stuff: Thing and Stuff Classes in Context. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1209–1218.
91. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 1137–1149. [CrossRef]
92. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–3 November 2019; pp. 9627–9636.
93. Strigl, D.; Kofler, K.; Podlipnig, S. Performance and Scalability of GPU-Based Convolutional Neural Networks. In Proceedings of the 18th Euromicro Conference on Parallel, Distributed and Network-Based Processing, Pisa, Italy, 17–19 February 2010; pp. 317–324.
94. Kim, W.; Seok, J. Indoor Semantic Segmentation for Robot Navigating on Mobile. In Proceedings of the 10th International Conference on Ubiquitous and Future Networks, Prague, Czech Republic, 3–6 July 2018; pp. 22–25.
95. Asadi, K.; Chen, P.; Han, K.; Wu, T.; Lobaton, E. LNSNet: Lightweight Navigable Space Segmentation for Autonomous Robots on Construction Sites. *Data* **2019**, *4*, 40. [CrossRef]
96. Han, S.; Mao, H.; Dally, W.J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv* **2015**, arXiv:1510.00149.
97. Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; Kautz, J. Pruning Convolutional Neural Networks for Resource Efficient Inference. *arXiv* **2016**, arXiv:1611.06440.
98. Anwar, S.; Hwang, K.; Sung, W. Structured Pruning of Deep Convolutional Neural Networks. *ACM J. Emerg. Technol. Comput. Syst.* **2017**, *13*, 1–18. [CrossRef]
99. Tremblay, J.; To, T.; Birchfield, S. Falling Things: A Synthetic Dataset for 3D Object Detection and Pose Estimation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2038–2041.
100. Jalal, M.; Spjut, J.; Boudaoud, B.; Betke, M. SIDOD: A Synthetic Image Dataset for 3D Object Pose Recognition with Distractors. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019; pp. 475–477.
101. Tan, W.; Qin, N.; Ma, L.; Li, Y.; Du, J.; Cai, G.; Yang, K.; Li, J. Toronto-3D: A Large-Scale Mobile LiDAR Dataset for Semantic Segmentation of Urban Roadways. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 202–203.
102. Chen, T.; Dai, B.; Wang, R.; Liu, D.; Chen, T.; Dai, B.; Liu, D.; Dai, B.; Liu, D.; Wang, R. Gaussian-Process-Based Real-Time Ground Segmentation for Autonomous Land Vehicles. *J. Intell. Robot. Syst.* **2014**, *76*, 563–582. [CrossRef]
103. Sun, L.; Yang, K.; Hu, X.; Hu, W.; Wang, K. Real-Time Fusion Network for RGB-D Semantic Segmentation Incorporating Unexpected Obstacle Detection for Road-Driving Images. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5558–5565. [CrossRef]
104. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; Sang, N. BiSeNet V2: Bilateral Network with Guided Aggregation for Real-Time Semantic Segmentation. *Int. J. Comput. Vis.* **2021**, *129*, 3051–3068. [CrossRef]