RESEARCH ARTICLE OPEN ACCESS

Road and Surrounding Detection and Segmentation Using U-Net Architecture

Raghava SaiNikhil [1], Dr.S.Govinda Rao [2]
Department of Computer Science and Engineering
Gokaraju Rangaraju Institutte Of Engineering and Technology
Hyderabad – India

ABSTRACT

The ability to understand the road and traffic layout is necessary to implement vision-based autonomous driving. The process involves detecting and classifying images of roads, pedestrians, vehicles, etc. Additionally, driving videos can be used to track different patterns of motion based on their spatial location. Temporal connections between lines show the arrangement of roads and their surroundings. For an accurate understanding of the road profile, we need to identify its various areas; roads, roadside, lane marks, vehicles, etc., make up the road profile in real-time. This problem can be solved by segmenting the different objects into different classes. By categorizing each pixel in an image, image segmentation is accomplished. Computer vision includes the field of image segmentation. Due to the development of neural networks (NN) and convolutional neural networks (CNN) over the past 10 years, computer vision problems have led to increased benchmarking results. The topic of semantic segmentation has been addressed by a number of DCNNs, including U-Net, RefineNet, DeepUNet, Inception V3, ResUNet, Inception V2, DenseNet, etc. Our paper presents a systematic review of different architectures and image processing techniques that have been proposed for semantic segmentation. Also included the datasets and model architectures used by various researches in this paper.

Keywords*:* - RefineNet, DeepUNet, Inception V3, ResUNet, Inception V2, DenseNet, Semantic Segmentation, Instance Segmentation

I. INTRODUCTION

Among the biggest challenges facing the automotive industry today is the development of autonomous driving. Autonomous cars that are reliable and affordable for the everyday consumer would have a revolutionary impact on society. In particular, it is expected that the number of road traffic fatalities will be reduced, traffic congestion will be reduced, fuel consumption will be reduced and noxious emissions will be reduced, and comfort for drivers will be improved, as well as expanding mobility for seniors and people with disabilities. As a result, every government and industry has been striving to make our mobility systems as autonomous as possible during the first decade of the new century. The aim of an autonomous vehicle is to perceive its immediate environment, act accordingly, drive accordingly to reach its desired destination in a safe and timely manner. This research paper focuses specifically on perception.

For humans to perceive their environment, they rely on their vision. It should however be noted that, in recent attempts to automate driving, most notably the DARPA Urban Challenge [6] and the demonstrations of autonomous driving by Google Inc. During the perception stage of the system, more reliance is placed on an extensive array of active sensors, such as lidars and radars that are employed to create 3D representations of the situation, radars, and lidars that are employed in order to detect obstacles in dynamic situations, or differential GPS to find the position of the vehicle within the previously generated 2D road maps. These sensors have a tendency to provide 3D (geometry) routes in a texture less format that can be used by the plane to plan a navigation route. This planning, however, will require semantic information as well, such as information about obstacles and their characteristics, such as their trajectories, speed, etc., and determining the presence of regulatory elements, such as semaphores, traffic signs, so that traffic rules like stopping speeds are respected. Computer vision algorithms are used to extract this semantic information from cameras. It has therefore been seen as practical to include several cameras on the prototypes of autonomous vehicles, even stereo rigs for augmenting the 3D imagery provided by active sensors.

An image segmentation program can either classify pixels according to semantic labels describing what they represent (semantic segmentation) or partition them according to individual objects (instance segmentation). This is a more difficult endeavor than image classification, which predicts a single label for each pixel, because semantic segmentation performs pixel-level labeling based on categories (e.g., human, car, tree, sky) for the whole image. Semantic segmentation with instance segmentation is extended to detect and delineate each object of interest in the image (e.g., a person based on his or her appearance in the image).

In this survey you will find an overview of the latest research in image segmentation science and an analysis of different types of deep learning-based segmentation techniques proposed to be used in the near future. This review provides an overview and insight into the different aspects of each method. Section 2 provides an overview of data resources we can use in segmenting datasets, followed by a literature review in section 3, which concludes the whole paper by including a discussion of this paper's theoretical framework. In section 4, we describe the performance metrics that are accompanied by a discussion on the conclusion and future scope of the project.

I. IMAGE SEGMENTATION PROCESS AND DATASETS

It isn't so difficult to segment data. Approximately five steps are involved, but they are divided into three modules as shown in Figure 2. The three modules are: Pre-processing, Processing, and Output. Rendering is a component of the model that inputs a series of images. During the pre-processing phase, we improve the contrast, brightness, color scheme, etc. of the images that may come in different sizes,

irregular brightness, etc. The processing phase is where the augmentation of the images, as well as training and validation of the model is accomplished. In the final outcome, masks and segments of the image can be seen.

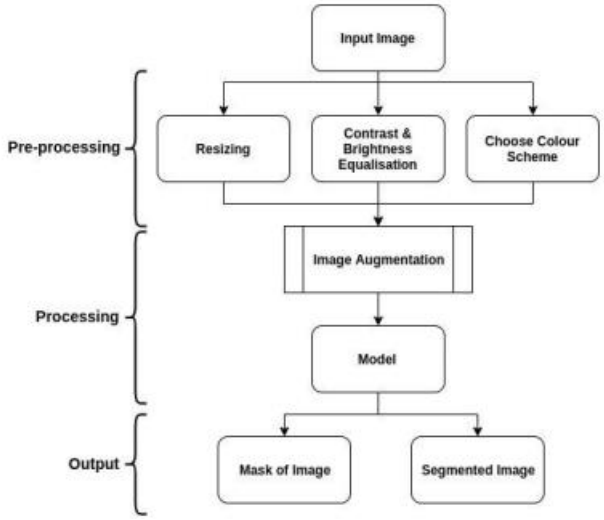

Fig. 1. Flowchart of basic image segmentation process

A. *Preprocessing*

If the input image consists of low exposure, brightness, contrast, or if it contains any anomalies, such as bad pixel alignment, then the image is considered a poor image. A poor image cannot be used to train the model. A model must have good quality input if it is to be trained to its maximum capacity. Images can be resized to make feature extraction more efficient. Images with fewer features are extracted when they are small, while images with more features are extracted when they are large. So the size of an image is directly proportional to how many features are in it. As a default, the DCNN creates images in 256X256 pixel format. As a result, training over larger images is also significantly faster compared to training over smaller images [1]. It is therefore important to resize before training so it helps to determine if any adjustments are needed. It is also a good idea to equalize the contrast and brightness in the model so that it operates more efficiently. When the level of contrast is low or when the level of contrast is high, the images are very difficult to learn since their characteristics like Texture, Edges, Color, etc., tend to be extremely unclear. For example, camera quality, environment conditions, background, etc., are all factors that affect the contrast and brightness of images. Oftentimes, the quality of the images in public datasets is satisfactory but in some instances, the task of improving the quality of the images can be challenging. There are different ways to preprocess an image so that you can achieve detailed features. For instance preprocessing plays an important role in the medical field. In order to store images in the real world, a diverse range of schemes are available, including grayscale, RGBA, RGB, CYN, and many more. Various color schemes even have different encodings and look different from each other. It is necessary to convert images into appropriate formats before training, for example the RGB format for 3 channels should be “256x256x3”, the BG format for 1 channel should be “256x256x2”, and RGBA for 4 channels should be “256x256x4”. There is a difference between the effects of different color schemes on model training [2].

B. *Processing*

During the training of the deep learning model, the model has to be trained on a vast amount of data. For instance, 3,000 images are required. A model that can be trained with high accuracy is very difficult with such little data. The dataset needs to be augmented significantly in order to increase accuracy [3]. A significant increase in images is achieved through rotation, shifting, cropping, and so on. This allows us to make use of the dataset more and more. Smaller datasets enable efficient and effective training of the model. It is necessary to use the original image as well as its mask in order to train the model. Each semantic model has its own strengths, and a few examples are U-Net, refined-net, deep-lab, etc. These models were trained for specific purposes, but they each have their strengths. For example, U-Net has been trained for biomedical images and it gives an accuracy of 77.5%. There are many different types of semantic segmentation networks. A challenge lies in choosing the right model before predicting the output.

C. *Output*

An image containing different pixels is predicted to contain different masks by a model. A specific segment of the image is created by XORing a specific mask with the original image. Accordingly, the application takes different decisions based on the predicted output classes.

II. LITERATURE REVIEW

CNNs have many different architectures, approaches, and algorithms for image segmentation. A CNN can be used in many different fields, and they each have different pros and cons. It is possible for one network to perform well in one area but not in another. For example, the Pyramid Scene Parsing Network (PSPN) is used to resolve outdoor scenes. In other words, architecture is made and trained to solve particular types of images [4]. As a result of this method, different classes of objects in the scene can be categorized, such as cars, buses, buildings, and backgrounds. Detailed descriptions of various CNNs with their underlying datasets and performance are discussed and compared.

In 2021 Zhixiong Nan et al., Developed a cross-attentional and inner-attentional object detection and semantic segmentation model [5]. In order to fully exploit the correlation between the two subdividing branches of detection and segmentation, the cross-attention mechanism enables the development of the essential interaction between them. Further, inner attention contributes to a better representation of a feature map's characteristics in the model by helping to strengthen its representation. Using a series of encoder-decoder networks, initial feature maps are extracted from images by first using an encoder-decoder network. It is then augmented with the feature maps to obtain segmentation feature maps, which are subsequently used to produce segmentation feature maps. In the next step, the cross-attention mechanism uses the object detection feature maps generated from the segmentation feature maps to guide it in the process of creating the segmentation feature maps [5]. Finally, the authors performed semantic

segmentation on the feature maps of segmentation followed by object detection on the detection feature maps. To evaluate this model, two popular public traffic datasets were analyzed. The present paper also conducted some ablation studies to ensure that the proposed inner attention and cross attention mechanisms are, indeed, effective, and the results of the study validate this claim. This claim is based on experiments conducted on the well-known dataset called Cityscapes [5]. The Cityscapes dataset is a set of image segmentation datasets collected from traffic scenes, which included 20,000 coarse-annotated images and 5,000 high-quality images. In this experiment, the dataset is constructed from 5,000 images with high-quality annotations, similar to those used in some existing studies, including DspNet. In order to separate 2,975 images for training and 500 images for testing from the 3,475 images in the training set, the annotations of the images in the testing set are not provided. A comparative analysis of the proposed model with the DspNet, BlitzNet, PairNet, and TripleNet models is performed using the Cityscapes dataset. The proposed model achieved detected and segmented the road, sidewalk, building, wall, etc., with 91.4%, 67.8%, 82.5%, 38.9% respectively. The Mean intersection over union (MIoU) of the proposed model is 57.4% which is higher compared with several other recently proposed methods, and this one achieves the best results. The comparison results are mentioned in table [5].

Table 1: Results of Zhixiong Nan et al., Developed a cross-attentional and inner-attentional object detection and semantic segmentation model

Method	road	swalk	build	wall	terrain
DspNet [4]	89.8	63.2	80.1	38.4	49.2
BlitzNet [12]	88.4	58.2	78.1	30.7	41.5
PairNet [5]	87.4	58.9	77.1	39.2	44.5
TripleNet [5]	87.7	60.6	77.7	38.3	45.8
Peng et al. [6]	90.7	65.0	81.0	**45.3**	48.1
Ours	**91.4**	**67.8**	**82.5**	38.9	**50.2**
–	sky	person	rider	car	**mIoU**
DspNet [4]	82.0	51.0	32.0	86.0	54.5
BlitzNet [12]	82.9	50.3	26.1	85.2	50.5
PairNet [5]	79.3	48.1	29.8	83.5	50.4
TripleNet [5]	80.5	49.1	27.8	84.8	51.3
Peng et al. [6]	83.8	53.5	33.1	86.4	55.5
Ours	**87.7**	**56.6**	**35.6**	**88.1**	**57.4**

In 2021, Ozan Unal et al., proposed an auxiliary 3D object detection task is explicitly leveraged as localization features in a novel Detection Aware 3D Semantic Segmentation framework (DASS) [6]. Through multitask training, the network's shared features are guided to be aware of per class detection features by which geometrically similar classes can be differentiated.

The authors also demonstrate how the added supervisory signal improves 3D orientation estimation capabilities by using DASS to generate high recall proposals for existing 2-stage detectors. There are two partial datasets used in this pipeline: (1) semantic labels for pointwise labels and (2) 3D object annotations for pointwise labels. Segmentation datasets are cropped to avoid introducing additional domain shifts since the detection datasets only provide annotations for the image FOV. Training for the 3D semantic segmentation task and the auxiliary 3D proposal generation task is performed using PointNet++ feature extractors trained on supervisory signals. An overview of network extensions. For PointRCNN, the RPN is DASS. Before generating a proposal, semantic feature fusion (SFF) is used to improve the results.Using the PointNet++ encoder-decoder, four set abstraction layers with multi-scale grouping are used in this architecture [6]. In each of the two scales, three linear layers follow the set abstraction layers' grouping and sampling operations [6].To obtain per point feature vectors rich in semantic and class-specific information, 4 feature propagation layers with skip connections are fed into the set abstraction layers. We can introduce scale invariance to our network by using 2-scale grouping, however, the hierarchical structure of the PointNet++ feature extractor captures more local properties that are advantageous to both tasks. Object proposal generation and 3D semantic segmentation heads use the same 1D convolution layer of size 128. Every layer is activated by batch norms and ReLUs. There is a learning rate of 0.002 [6]. A one-cycle learning rate scheme is used with Adam optimizer. Default values for weight decay and momentum are 0.001 and 0.9, respectively. These experimentations done using SemanticKITTI and KITTI object datasets and achieved good results which are shown below.

Table 2: Results of Ozan Unal et al., proposed an auxiliary 3D object detection

	BEV [%]			Orientation [%]		
Method	Easy	Moderate	Hard	Easy	Moderate	Hard
PointRCNN[30]	**92.13**	**87.39**	**82.72**	95.90	91.77	86.92
DASS+RCNN	91.74	85.85	80.97	**96.20**	**92.25**	**87.26**

In 2020 Jianbo Liu et al., proposed a holistically-guided decoder model to obtain high-resolution semantic-rich feature maps via the multi-scale features from the encoder for semantic segmentation [7]. By combining high-level and low-level features from the encoder, we create a novel, holistic codeword generation and assembly operation that allows for decoding. Accordingly, the researchers have implemented the Efficient FCN architecture as a way of semantically segmenting data and the HGD-FPN architecture as a method of object detection and segmentation of instances for the proposed holistically-guided decoder. With only 1/3 of the computation cost of state-of-the-art methods, the Efficient FCN achieves comparable or better performance for semantic segmentation on PASCAL VOC, PASCAL Context, and MS COCO datasets [7]. Using this HGD method, we achieve high resolution features by combining multiple scales of features, providing holistic codewords and assembling the codewords for high resolution up-sampling [7]. 1) One approach that was used in the study was to utilize the multi-scale feature maps generated by the encoder in order to generate different types of holistic codewords that encoded different aspects of the input image's global context, and linear assembly coefficients to achieve high-resolution feature maps based on the holistic codewords. Assuming that there are three feature maps generated by encoders of different sizes (OS=8, 16 and 32) and that the decoder develops semantic-rich feature maps of scale OS=8, authors assumed that the encoders could

generate three feature maps at different sizes. To denote the concatenation along the channel dimension, authors used this to represent bilinear down-sampling. 2) The n holistic codewords are generated via a combined multi-scale feature map (m32). Then using two separate 1 * 1 convolutions, a codeword base map and n spatial weighting maps are generated. The technique uses spatial weighting bases from all spatial locations of the world to learn to linearly combine these into a single codeword that encompasses some aspect of the global context. There are ultimately n weighting maps to describe a holistic way of encoding global features that are of high level. (3) A new subspace-level metric for high-resolution feature up-sampling is also proposed called codeword assembly. By using the multi-scale multi-scale fused features m8, the authors deduced the linear assembly coefficients of the n codewords based on spatial locations to generate an interactive high-resolution feature map. Semantic segmentation was achieved by combining EfficientFCN and HGD. As a reference, the dilated FCN methods apply the dilated convolutional algorithm with dilation rates 0, 2, and 4 rather than the stride employed by the backbone networks, thus removing the last two strides of the backbone networks. The original ResNet is used as the encoder's backbone network. In this case, as a result of this ResBlock, the output feature map is 32×32 smaller than that of the input image. Using the proposed holistic-guided decoder, authors feed the encoder feature maps into the decoding algorithm, which then uses the output upsampled feature map, *f8, to perform the classification process. For the initialization of the encoder network, the pre-trained weights from ImageNet are utilized. Using the proposed EfficientFCN model on the PASCAL VOC 2012 test set, 85.4 % of the test sets were obtained without using the MS COCO dataset pre-training and 87.6% with the MS COCO dataset pre-training. These results are reported in the following table below [7].

Table 3: Results of Jianbo Liu et al., proposed a holistically-guided decoder model

Method	aero	bike	bird	boat	bottle	mIoU%
FCN [42]	76.8	34.2	68.9	49.4	60.3	62.2
DeepLabv2 [1]	84.4	54.5	81.5	63.6	65.9	71.6
CRF-RNN [43]	87.5	39.0	79.7	64.2	68.3	72.0
DeconvNet [44]	89.9	39.3	79.7	63.9	68.2	72.5
DPN [45]	87.7	59.4	78.4	64.9	70.3	74.1
Piecewise [46]	90.6	37.6	80.0	67.8	74.4	75.3
ResNet38 [47]	94.4	72.9	94.9	68.8	78.4	82.5
PSPNet [19]	91.8	71.9	94.7	71.2	75.8	82.6
EncNet [5]	94.1	69.2	**96.3**	76.7	**86.2**	82.9
APCNet [6]	95.8	75.8	84.5	76.0	80.6	84.2
CFNet [7]	95.7	71.9	95.0	76.3	82.8	84.2
DMNet [18]	96.1	**77.3**	94.1	72.8	78.1	84.4
Ours	**96.4**	74.1	92.8	75.6	81.9	**85.4**
CRF-RNN [43]	90.4	55.3	88.7	68.4	69.8	74.7
Piecewise [46]	94.1	40.7	84.1	67.8	75.9	78.0
DeepLabv2 [1]	92.6	60.4	91.6	63.4	76.3	79.7
RefineNet [38]	95.0	73.2	93.5	78.1	84.8	84.2
ResNet38 [47]	96.2	75.2	95.4	74.4	81.7	84.9
PSPNet [19]	95.8	72.7	95.0	78.9	84.4	85.4
DeepLabv3 [2]	96.4	76.6	92.7	77.8	87.6	85.7
EncNet [5]	95.3	76.9	94.2	80.2	85.2	85.9
CFNet [7]	**96.7**	79.7	94.3	78.4	83.0	87.2
Ours	96.6	**80.6**	**96.1**	**82.3**	**87.8**	**87.6**

In 2020, Youngwan Lee et al., proposed a simple and efficient anchor-free instance segmentation, called Center Mask, which adds a novel spatial attention-guided mask (SAG-Mask) branch to anchor free one stage object detector (FCOS) in the same vein as Mask R-CNN [8]. FCOS object detector and the SAG-MASK branch are used together with the spatial attention map in this proposed system. The spatial attention map helps to focus on informative pixels and suppress noise by detecting and predicting segmentation masks on each detected box. Also, the authors propose an improved backbone network, VoVNetV2, that introduces two new useful strategies: (1) residual connections to alleviate some of the problems associated with optimization in larger VoVNets, and (2) effective squeeze excitation (eSE) to deal with the issue of channel information loss associated with original squeeze excitation. Researchers created SAG-Mask and VoVNetV2 to target large- and small-sized models, respectively, with CenterMask and Center Mask-Lite. Taking the input scale into account for scale-adaptive ROI assignment, the APmask improved by 0.4% as per their ablation study. In its initial state, the FCOS detector starts with 37.8% APbox and 57 ms of running time [8]. By adding only a naive masking head, the APbox performance improves by 0.5% and 33.4% mask is obtained [8]. SAM, the spatial attention module, makes the mask performance forward, because it enhances the mask prediction by focusing on informative pixels while also suppressing the noise because authors already discussed a scale-adaptive RoI map strategy previously. In the later stage, researchers recalibrated IoU mask prediction, resulting in an improved APmask performance score of 0.7%. In the second phase, researchers extracted the features using FCOS detectors [8]. As a result, they upgraded to VoVNetV2, which is based on residual connections and uses effective squeeze-excitation (eSE) to make a more diverse network mesh. A validation of eSE was carried out by applying SE to the VoNet and comparing it with eSE. Compared to VoVNetV1, the proposed eSE functions in a similar way to the SE only it uses one FC layer to maintain channel information and boosts the APmask and APbox. These backbone models were compared to 11 other backbone models including MobileNet V2, VoVNetV2-19/39/57/99, HRNetV2-W18/W32/W48, and ResNet-50/101/Xt101, which were all trained using 12 epochs with 16 batches using the COCO va12017 dataset without augmenting them. The segmentation mask's performance is compared in the table below with comparable speed [8].

Table 4: Youngwan Lee et al., proposed a simple and efficient anchor-free instance segmentation

Backbone	Params.	AP^{mask}	AP^{box}	Time (ms)
MobileNetV2 [31]	28.7M	29.5	32.6	**56**
VoVNetV2-19 [19]	37.6M	**32.2**	**35.9**	59
HRNetV2-W18 [32]	36.4M	33.0	36.7	80
ResNet-50 [10]	51.2M	34.7	38.8	72
VoVNetV1-39 [19]	49.0M	35.3	39.7	**68**
VoVNetV2-39	52.6M	**35.6**	**40.0**	70
HRNetV2-W32 [32]	56.2M	36.2	40.6	95
ResNet-101[10]	70.1M	36.0	40.7	91
VoVNetV1-57 [19]	63.0M	36.1	40.8	**74**
VoVNetV2-57	68.9M	**36.6**	**41.5**	76
HRNetV2-W48 [32]	92.3M	38.1	43.0	126
ResNeXt-101 [36]	114.3M	38.3	43.1	157
VoVNetV1-99 [19]	83.6M	31.5	35.3	**101**
VoVNetV2-99	96.9M	**38.3**	**43.5**	106

In the year 2020, Diakogiannis et al. developed a novel image segmentation method. Their research focused on aerial images, which are used to monitor remote areas for

intelligence purposes [9]. Deep learning was a key factor behind the development of remote sensing in this research. As per their research, pixel by pixel segmentation is useful for monitoring and analyzing a vast range of fields such as agriculture, military, forecasting, etc. In total, there are three layers in the system: A matrix U-Net serving as the backbone, residual blocks for detecting exploding and vanishing gradients, and Pyramid Scene Parsing Pooling (PSPP) for background content decoding. ResU-Net has 31 layers in total, including six convolution layers, ten residual layers, five up-sampling layers, six combine layers, and two PSPP layers. State-of-the-Art models were used in the study, such as U-Net and PSPP. U-Net architecture and skip connections form the foundation of ResU-Net. By using skip connections, up blocks can access similar features in down blocks. This results in bottlenecks in downstream layers. The network may not be able to learn and in the worst-case scenario, it may create a dumb network [9]. An exploding gradient or vanishing gradient can be resolved with residual blocks. The ResU-Net is built on residual blocks. It is caused by the fact that a network may have too many layers, causing weights to change exponentially or the activation functions to function erratically. It was for this reason that ResU-Net introduced three parallel atrous convolution layers with a kernel size of 3 and a stride of 1 aimed at addressing this issue. Each layer has been given atrous layers that increase its receptiveness [9]. ResU-Net consists of two parallel matching methods, designed by Zhao et al., in 2017, for scene parsing and pooling, which make up the core architecture of the system. It is used as a PSPP layer to connect the down and up blocks in the middle and another layer at the end of the architecture. It is primarily used to remove background noise. Four parallel pools are contained within this layer. It consists of four blocks, namely Input, Max pooling, Restore DIM, and Convolution. In order to evaluate and perform ResU-Net, the dataset created by ISPRS2D Potsdam in 2018 was utilized for evaluation. There are six classes of images included in the report including building, surface, car, tree, low vegetation, and background images. The results from ResU-Net are Average F1 score 92.9%, Accuracy 91.5%, and Best Accuracy 91.5% [9].

Deep U-net (Yang, L et al., 2018) was developed in 2018 for Sea-land image segmentation [10]. There are 207 hand-crafted labeled images in the dataset constructed by the author. The author also uses photos that have been processed by Google. This dataset includes some images with noise. Segmenting these images with binary models is a very difficult task. Satellite images of the sea and land are widely used for remote sensing. Marine and terrestrial areas can be distinguished on satellite images. Using their algorithms, ROIs are segmented by thresholds, shapes, edges, etc. To find out how U-Net and SegNet connected with the sea-land dataset, their interactions were analyzed. First, the U-Net segmentation model and SegNet segmentation model were developed to build deep learning segmentation models [10]. In general, Deep U-Net differs from U-Net primarily through its blocks and connections. An up-sampling block is used by Deep U-Net, as well as a down-sampling block. Deep U-Net also uses two connections, namely the "U" connection and the "+" connection. The "U" connection enables operation on a group of data by connecting the up-block and down-sampling block together. A Plus connection allows a group of adjacent blocks to operate together as one. It is important to note that the down-sampling block is comprised of two convolutional layers connected by the ReLU. First and second layers of the convolution system use three times three or 3X3 convolution layers respectively with a 32 and 64 kernel size [10]. Most often, two smaller convolutional layers are substituted for the large convolutional layer in deep learning algorithms. As the input, convolution takes the output of each layer and passes it along to the layer that will pool the data and do the up-sampling. In addition, the output of down-sampling is concatenated with the start of convolution in the up-sampling block. Using the same maximum entropy method, authors calculated a total layer count of 28 layers, having 14 convolution layers, seven pooling layers, and seven up-sampling layers. Several parameters were compared between Deep U-Net and U-Net in order to find out which is best: F1 score, land precision, land recall, overall precision, and overall recall. With its F1 score of 4.8% above U-Net and 1.92 percent above SegNet, Deep U-Net scored higher than the two other competitors [10].

Using pixel-wise person segmentation with other DCNNs, Lin et al., in 2017, presented a quandary [11]. The problem with semantic segmentation networks that existed before Refine-Net is the same problem that they all face. In addition to too many convolutions and pooling, they use too many other techniques. Layers cause the image to shrink and the texture map to be blurred. This problem has been addressed by the Refine-Net: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation of 2016 paper [11]. Multipath refinement is introduced at each stage of the proposed method. Down-sampling aids in learning at the essential level or upper layer by exploiting the learning. Through long-range residual connections, higher semantic features are summed up to finer details. Three modules make up Refine-Net: Multiresolution Fusion, Residual Convolution Units, and Chained Residual Pooling. They combine to achieve end-to-end prediction. A deep neural network layer called RCU has the capability of fine-tuning features maps using sequential convolution. This model collects background information, removes noise, etc., using a chained residual pooling module. A separate background from a foreground module is also implemented. Multi-resolution fusion is used to combine all features. ADE20K MIT for scene parsing was used in conjunction with SUN-RGBD, PASCAL VOC 2012, NYUDv2, Cityscapes, PASCAL-Context, and PASCAL-Context for scene parsing [11]. Parsing people is done using the person-part dataset. Parsing people is done with the person-part dataset. Refine-Net came up with an IoU of 83.4% [11].

Full-resolution residual networks were proposed by Hermans et al., 2017 for semantic segmentation in street scenes [12]. The network is used to segment streets and classifies them according to their type, such as Bus, Car, Building, Background, etc. 5000 images of street scenes comprise the Cityscapes dataset, for which it's been trained and tested. Researchers used a state-of-the-art Res-Net model, which is a widely used classification method. They design architecture with two streams, one of which is a high-level specification that reaches the end of the network with an adequate resolution, and the other is a layer of convolution, pooling, etc. that carries information to the end. Those notes that carry information to the end of the stream are combined with the subsequent residuals at each level

[12]. Creating a distinction between high-level features and low-level features is therefore required to support the model. By pooling fine details, each layer creates edges, boundaries, etc. from granular information concatenated with it. Bootstrap and cross-entropy losses have been configured in the model. There are 2 types of augmentations used to reduce overfitting: Translation augmentation and Gamma augmentation. In order to train, validate, and test their models, the authors divide the dataset of 4998 images into 3 subsets of 2,974, 500, and 1,524 images. In the benchmarking of Cityscapes, they succeeded in producing the best IoU of 71.80% [12].

In this study, Ronneberger, et al. Proposed the implementation of deep neural networks as a means for pixel-level segmentation in 2015 [13]. Using data augment to utilize the available annotated samples more efficiently, they presented a network and training strategy. An expanding path provides precise localization, while a contracting path captures context. A contracting path was the winning method at ISBI 2012. By creating localization patches using convolutional neural networks, U-Net uses a similar concept. IEM segmentation challenge dataset is used in this study, which was initiated at ISBI 2012 and is still open for contributions. Training data consists of 30 images of the first instar larval ventral nerve cords (VNC) using serial section transmission electron microscopy [13]. An annotated ground truth segmentation map of the corresponding images is provided for each cell (white) and membrane (black) [13]. A threshold of 10 different levels is applied to the map, after which computations are performed for "warping error", "Rand error", and "pixel error". There are two paths to expand U-Net: ones for contracting and ones for Symantec. Specifically, a contraction is accomplished through the use of a convolution layer followed by ReLU. As the features map size gets doubled on each convolution layer, it is like 4,8,16, and 32. Maximizing the pooling of major features is achieved using 2X2 max-pooling after every convolution layer. Expanding blocks of features now sample up following contraction of features. A layer of up-convolution is used in the up-sampling block [13]. This is followed by 23 additional layers. It is at this point, that the output of the 1X1 convolution layer is determined based on how many labels are required. If there are only a few samples in the dataset, image augmentation is used to enhance the image. Image augmentation is used to train the U-Net. Additionally, they used the U-net for the task of segmenting cells within light microscopic images. As part of the ISBI's cell tracking challenges in 2014 and 2015, this segmentation task has been undertaken. In the sequence "PhC-U3732", the molecules were plated onto a polyacrylamide substrate on a Glioblastoma-astrocytoma U373 cell line and were observed by phase-contrast microscopy. Annotated training images of 35 images are included. Researchers were able to achieve a 92% intersection over union (IoU) using this dataset. With this algorithm, the best IoU for the second data set is 77.5% for DIC-HeLa [13].

III. COMPARISION BETWEEN DIFFERENT RESEARCH STUDIES

Researchers worked with existing datasets to compare their architectures, but these datasets were saturated. Building a large, vibrant data set is a challenging process as it requires a lot of effort and thought. The interconnection over union (IOU) has been argued to be an important parameter in comparison between different networks but it's also important to consider processing speed and training time in the comparison.

Table 5: Comparision Table

Year	***Architecture***	***Dataset***	***Accuracy***	***Ref. No***
2021	Joint object detection and semantic segmentation model with the cross-attention and inner-attention mechanisms	City Scapes	Road-91.4% SideWalk-67.8% Sky-87.7% Building-82.5% Car-88.1% and MIoU-57.4%	[5]
2021	Detection Aware 3D Semantic Segmentation (DASS) with PointNet++ (encoder decoder) and RCNN	SemanticKITTI and KITTI	Car: BEV-92.13% orientation- 96.2%	[6]
2020	EfficientFCN with Holistic- guided decoder	Pascal VOC2012 & MS COCO	87.6	[7]
2020	SAG-mask and FCOS with backbone architecture VoVNetV219/39/57 /99 layers to make CenterMASK-lite method	COCO dataset 2017	V-19 **AP^{mask}** - 32.4%. **AP^{box}** - 35.9% **and FPS**-43.5	[8]
2020	ResUNet	ISPRS 2D Potsdam Sea-Land DataSet	F1: 98.9%	[9]
2018	DeepUnet	Self made sea-land dataset from google earth images	F1: 95.39%	[10]
2017	RefineNet	PASCAL VOC 2012	83.40%	[11]
2017	Full-Resolution Residual Networks (FRRN)	Cityscapes benchmark	71.80%	[12]
2015	U-Net	DIC-HeLa segmentation challenge	77.50%	[13]

IV. CONCLUSION

Various DCNNs and image processing methods are presented in this paper for semantic segmentation. Computer vision has rapidly expanded in recent years. The computer vision field is tilting more towards CNNs due to the evolution of this technology. In nearly every field in which you find computers, semantic segmentation has become one of the most widely applied applications of computer vision. It includes three steps, i.e. preprocessing, processing, and presenting the output of the segmentation process.

Preprocessing is the process of making the image smaller, brighter, more contrast, etc. There is also augmentation that takes place at this stage, comprising of choosing the model. After choosing the model, there are outputs of mask images as well as segmented images. A study has been carried out in this paper to explore DCNNs in their database and to examine the applications which are possible on the basis of these DCNNs. We analyze their working and applicability by examining their accuracy as well as their efficiency. Observing an extensive review of a number of papers, the present research shows that even though various DCNNs are proposed for different datasets, there is still a need to build a robust architecture with better real-time performance.

V. REFERENCES

[1] "Pal, Nikhil R., and Sankar K. Pal. "A review on image segmentation techniques." Pattern recognition 26.9 (1993): 1277-1294.".

[2] "Plataniotis, Konstantinos, and Anastasios N. Venetsanopoulos. Color image processing and applications. Springer Science & Business Media, 2000.".

[3] "Truong, Thanh-Nghia, Vu-Duy Dam, and Thanh-Sach Le. "Medical images sequence normalization and augmentation: improve liver tumor segmentation from small dataset." 2018 3rd International Conference on Control, Robotics and Cybernetics (CRC). IEEE, 2018.".

[4] "Zhao, Hengshuang, et al. "Pyramid scene parsing network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.".

[5] "Nan, Zhixiong & Peng, Jizhi & Jiang, Jingjing & Chen, Hui & Yang, Ben & Xin, Jingmin & Zheng, Nanning. (2021). A Joint Object Detection and?Semantic Segmentation?Model with Cross-Attention and Inner-Attention Mechanisms. Neurocomputing. 463. 10.1016/j.neu".

[6] "Unal, O., Van Gool, L., & Dai, D. (2021). Improving point cloud semantic segmentation by learning 3d object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2950-2959).".

[7] "J. Liu, J. He, Y. Zheng, S. Yi, X. Wang and H. Li, "A Holistically-Guided Decoder for Deep Representation Learning with Applications to Semantic Segmentation and Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10".

[8] "Lee, Y., & Park, J. (2020). Centermask: Real-time anchor-free instance segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 13906-13915).".

[9] "Diakogiannis, F. I., Waldner, F., Caccetta, P., & Wu, C. (2020). ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. ISPRS Journal of Photogrammetry and Remote Sensing, 162, 94-114.".

[10] "R. Li et al., "DeepUNet: A Deep Fully Convolutional Network for Pixel-Level Sea-Land Segmentation," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 11, pp. 3954-3962, Nov. 2018, doi: 10.1109/JSTARS.2018.28".

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

[12] "Pohlen, Tobias, Alexander Hermans, Markus Mathias, and Bastian Leibe. "Full-resolution residual networks for semantic segmentation in street scenes." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4151-4160. 2017.".

[13] "Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.".