Scale Invariant Semantic Segmentation with RGB-D Fusion

Mohammad Dawud Ansari[1], Alwi Husada[2] and Didier Stricker[1]

Abstract— **In this paper, we propose a neural network architecture for scale-invariant semantic segmentation using RGB-D images. Along with color images, we incorporate depth images as an additional input modality. Especially in an outdoor scene which consists of objects having different scale caused due to their distance from the camera. The near distance object consists of significantly more pixels than the far ones. We adapt to a well-known DeepLab-v2(ResNet-101) for semantic segmentation model as our RGB baseline. A separate depth branch takes depth image as input. Later, the intermediate feature maps generated from both color and depth image branches are fused using a novel fusion block. The global context is considered using a global average branch skipping the pyramid pooling network. The results obtained are comparable to the state-of-the-art. Our model is compact and can be easily applied to the other RGB model. We perform extensive qualitative and quantitative evaluation on a challenging dataset Cityscapes. Additionally, we evaluated our model on a self-recorded real dataset. For the sake of extended evaluation of a driving scene with ground truth, we generated a synthetic dataset using popular vehicle simulation project CARLA. The results obtained from the real and synthetic dataset shows the effectiveness of our approach.**

I. INTRODUCTION

Deep Convolutional Neural Networks (DCNNs) have shown remarkable accuracy for computer vision tasks like object classification [5]–[8] and object detection [9]–[13]. In the era of autonomous navigation, classification and detection are not sufficient information that can guide a vehicle in an unknown environment autonomously. As an example, such a system must be able to detect pedestrian from car and house and so on. This is also useful for robotic vision and medical data imaging [14].

Dense classification to assign the same label to pixels that belong to an object in the 2D image is called as pixel-wise semantic segmentation. Earlier approaches [2]–[4], [15], [16] provide reasonable accuracy. However, this task is difficult with implicit problems such as illuminations, occlusions, cluttered background and multi-scale object instances.

Earlier multi-scale problem is solved by employing a network which takes input images with multiple resolutions and later aggregates the feature maps [17]–[19]. Since the computation is performed parallelly for different scale, the overall computation cost is higher, in terms of memory requirement and computation power. Other alternatives to solve the multi-scale problem are proposed by [4], [16]. Depth information can help to solve the problem of ambiguity in the scale [3]. Additionally, contextual information can be obtained by parallel pyramid pooling network as proposed in [15]. We merge multi-scale feature generation along with contextual information and the depth information to tackle the problem of scale in driving scenes. We summarize the contributions of the paper as follows:

- We propose a new neural network architecture that takes depth images as additional input. In the network, the diminishing resolution of activation maps are maintained using dilated convolution. The intermediate feature maps from depth branch and RGB branch are merged using a novel fusion block. This depth information resolves the scale ambiguity in the image caused due to similar objects at different distances.
- We describe a pyramid pooling network which can cope with difficult scale changes for similar objects. Small dilation rate is used at every level of the pyramid pooling network which majorly focuses on the smaller objects. However, to maintain the invariance for bigger objects, we adapt Global Average Pooling for contextual information.
- We describe a generation and utilization of a synthetic dataset using vehicle simulation project CARLA [20]. An extensive quantitative evaluation is performed on Cityscapes [21] and synthetic driving dataset CARLA [20]. Additionally, we perform the qualitative evaluation on a self-generated real Zed dataset. The results obtained using our proposed model are promising for difficult scale changes of the similar object, which is also comparable to the state-of-the-art.

II. RELATED WORKS

Over the past few years, the breakthroughs of Deep Learning in images classification were quickly transferred into the semantic segmentation task. The introduction of Fully Convolutional Network [1] which modifies the last fully connected layer to spatial output, enables for solving pixel-wise semantic segmentation. The issues in the deep fully convolutional network is a set of stride in convolution and maximum/average pooling. These operations downsample the spatial size of feature maps, causing the dramatic reduction in the resolution which affects the final prediction. Several approaches have been proposed to remedy the issue. Shelhamer *et al.* [1] upsample the feature maps from top layers and concatenate it with the feature maps from intermediate layers. Eigen and Fergus [22] cascade multi-scale Deep Convolutional Neural Network (DCNN). They concatenate fine-details result with the coarse input and use it as an input to the next DCNN. Noh *et al.* [23], used an encoder-decoder architecture with deconvolution also known as transposed convolution in the decoder part.

{md_dawud.ansari, didier.stricker}@dfki.de and husada@rhrk.uni-kl.de
[1]Department of Augmented Vision, DFKI [2]University of Kaiserslautern

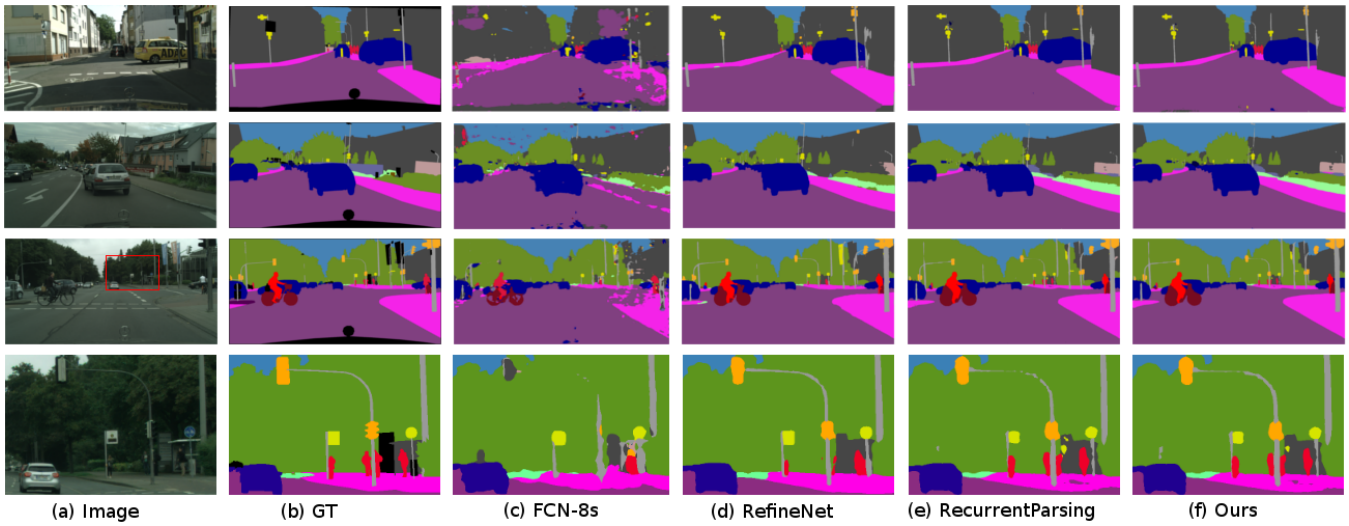

Fig. 1. Qualitative results of our model compared to FCN-8s [1], RefineNet [2], RecurrentParsing [3] and DeepLab-v2(ResNet-101)[4] on Cityscapes dataset (validation set).

Another approach is proposed by Chen *et al.* [4] referred as DeepLab-v2(ResNet-101), which we use as our baseline. They introduced "atrous convolution" to adjust receptive fields size by inserting "holes" in the filter. It expands the receptive field without adding extra parameters. Wang *et al.* [24] introduced Dense Upsampling Convolution (DUC) to upsample the feature maps which uses convolution operation instead of deconvolution or bilinear interpolation.

Several DCNN models are proposed which combines depth information with color information for semantic segmentation [25]–[27]. The fusion strategy affects the overall accuracy of the model. We experiment our model with different fusion strategy and use the optimal one. Hazirbas *et al.* [25] proposed an encoder-decoder type model that use two branches in encoder part to extract feature maps of RGB and depth images separately. They sum both feature maps which is later fed to the decoder part to get the final prediction. We adapt similar methodology in our approach with additional modifications. Cheng *et al.* [27] introduces a gated fusion method. It consists of concatenation, convolution and sigmoid layer. They concatenated the top feature maps from RGB and depth branch, and then the resulting features are convolved with 3×3 filters followed by sigmoid layer to regularize the features. The outputs of sigmoid are used to weigh the contribution of depth and RGB features. Park *et al.* [26] extends RefineNet [2] architecture to incorporate depth information known as Multi-modal Feature Fusion (MMF). Kong and Fowlkes [3] learned quantized depth for gating the network which is trained using Recurrent Neural Network's strategy. Ansari *et al.* [28] modified the same idea by proposing an adaptive pooling network. In our work, we aim to adjust the receptive field with adjacent depth network. The goal is achieved by fusing both pieces of information together during feature map generation. This makes the intermediate feature maps scale invariant.

III. PROPOSED ARCHITECTURE

The proposed model architecture is motivated from DeepLab-v2(ResNet-101) [4]. This model has two major parts. The first part is referred as ResNet-101 [29] for feature maps generation and the second part is referred as Pyramid Pooling. This combined network is named Atrous Spatial Pyramid Pooling (ASPP). Our model can be seen in Figure 2. In contrast to DeepLab-v2(ResNet-101) [4] our model has a parallel branch that accepts depth image as input. The features are generated from color as well as depth image separately. We then combine feature maps from both branches in the fusion block, which is passed to the ASPP and later to final prediction. The ASPP consists of five parallel convolutions + Batch Normalization + ReLU blocks with different dilation rate.

A. RGB-D Architecture

We will briefly review DeepLab-v2(ResNet-101) architecture proposed by [4]. One implicit problem in Deep Convolution Neural Network (DCNN) for Semantic Segmentation is consecutive maximum/average pooling and striding in convolutions operation that reduce spatial input dimension into significantly smaller feature maps. This downsample factor is about 32 with the actual input dimension. To overcome this issue Chen *et al.* [4] proposed atrous convolution also known as dilated convolution. The output signal $o\{i\}$ obtained after applying a dilated convolution in a 1D signal is given as:

$$o[i] = \sum_{l=1}^{L} x[i + r \cdot l] f[l], \tag{1}$$

where $x[i]$ is an input signal, $f[l]$ is an filter that has length L and r is a dilation rate. Note that, when $r = 1$ is a standard convolution. In a 2D convolutional operation, atrous convolution can be seen as inserting 'holes' to the convolution filter (inserting zeros between two neighboring

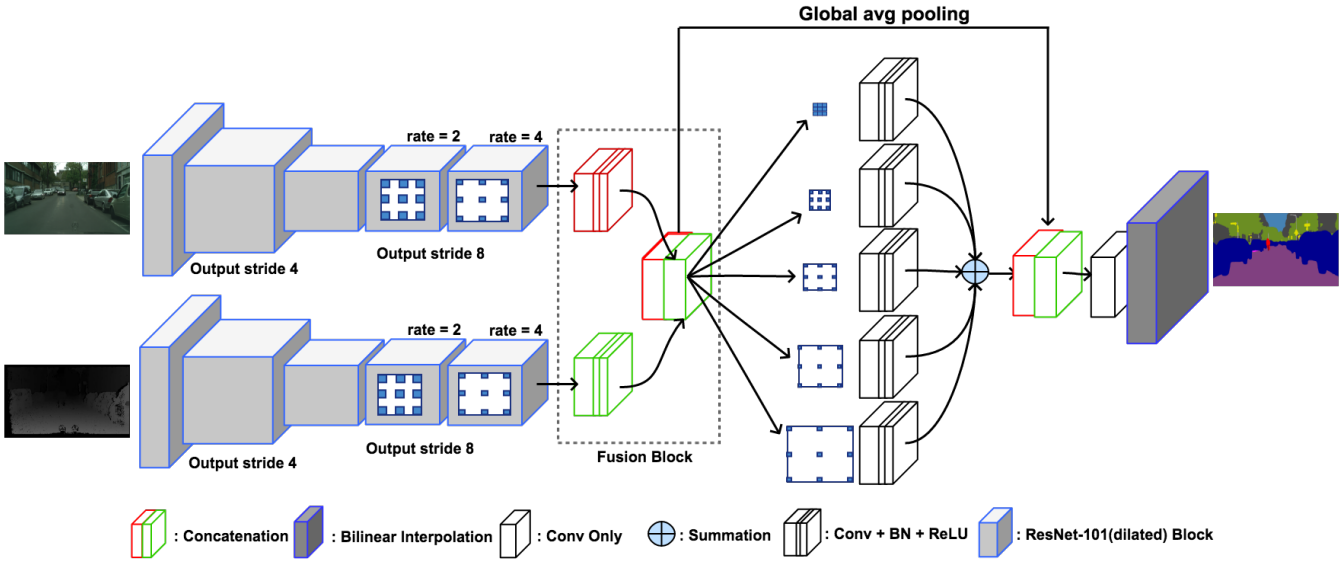

Fig. 2. Overview of our proposed architecture. The model mainly composed of two branches of features generation, Fusion Block Layer and Pyramid Pooling. We use Atrous ResNet-101 to extract robust features from both RGB and depth images. We then combine feature maps from both branches in the Fusion Block, which is passed to the Pyramid Pooling and later to final prediction. Additionally, Global Average Pooling is applied to get contextual information.

pixels in the filter). By defining the dilation rate r, it allows us to modify the size of receptive fields without changing the filter size. In addition, it also lets us control the spatial output size of feature maps of convolutions operation. In DeepLab-v3 [16] the spatial output size of feature maps is denoted as *output_stride*, which can also be termed as the reduction factor of the spatial input size to produce the desired output. For example, in original ResNet-101 model, the *output_stride* is 32 [29]. In order to obtain a bigger spatial dimension of the output feature maps, assuming the *output_stride* to be 16, the stride of the last pooling and convolution layer is set to 1 and modify the dilation rate of the subsequent convolution layers to 2. The feature maps from the last ResNet-101 layer are fed to ASPP. It consists of several parallel filters with different dilation rate to exploit different scale of features. In DeepLab-v2(ResNet-101) [4], the ASPP composed of four different filters with dilation rate $\{6, 12, 18, 24\}$. They are summed together before upsampled by *output_stride* factor with bilinear interpolation.

The repetitive convolution operation causes the diminishing of size of the features maps. An alternative to this operation is the use of *output_stride* $= 1$ along with dilation rate and modifying the stride of all the convolution layer to obtain same size feature maps. However, this operation is costly in terms of memory and training time. Thus *output_stride* $= 8$ is a reasonable choice to deal with the trade-off between memory usage and accuracy. Hence, we use *output_stride* $= 8$ in our model. To handle depth data, we add an additional dilated ResNet-101 [29] branch. We modify the first convolution layer to accept one channel image instead of three channels (RGB). The rest of the layers in depth branch are same as its RGB counterpart. The explanation of this part of the model is shown in Figure 2.

B. Fusion Block

One trivial way to integrate depth information into existing RGB-based model, is by stacking both images as a single 4-channel RGB-D image and modifying the first convolution layer to accept a 4-channel input [30]. Furthermore, Hazirbas *et al.* [25] showed that stacking RGB-D input produces less discriminant features than when fusing RGB and depth feature maps. The intuitive idea of fusing RGB and depth feature maps can be seen in the term of neuron-wise activation. For a given pixel, the activation maps from color and depth branch compliment each other simultaneously thereby producing more accurate segmentation. Motivated from this idea along with Hazirbas *et al.* [25] we propose fusion block which is composed by convolution followed by summation or concatenation. Given feature maps of RGB and depth from the last Atrous ResNet-101 with dimension $H \times W \times 2048$ where H and W are height and width of the feature maps, we reduce the dimension to be $H \times W \times 512$. We then fuse both feature maps to get more discriminant features representation, which are then fed to the pyramid pooling network. We experiment with sum and concatenation for fusing the feature maps. The results show that concatenation produce better accuracy (see Table I).

C. Pyramid Pooling

Many earlier approaches were proposed which provides significant accuracy with single object without much variation in the object's scale [1], [23], [25]. This creates an implicit issue for segmenting multi-scale objects in a scene. Especially small objects which are far away from the camera. Motivated from [4], [25], we intend to solve this issue by incorporating depth information and applying pyramid pooling. In contrast, we apply five parallel convolutions +

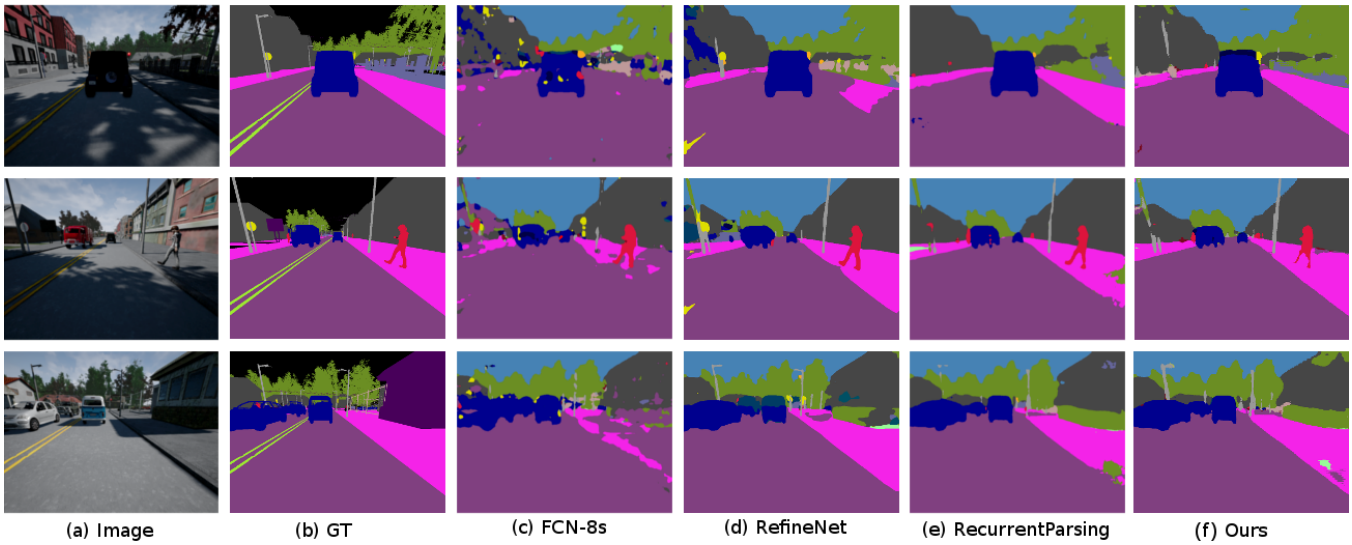

Fig. 3. Qualitative results of our model compared to FCN-8s [1], RefineNet [2], RecurrentParsing [3] and DeepLab-v2(ResNet-101)[4] on CARLA dataset.

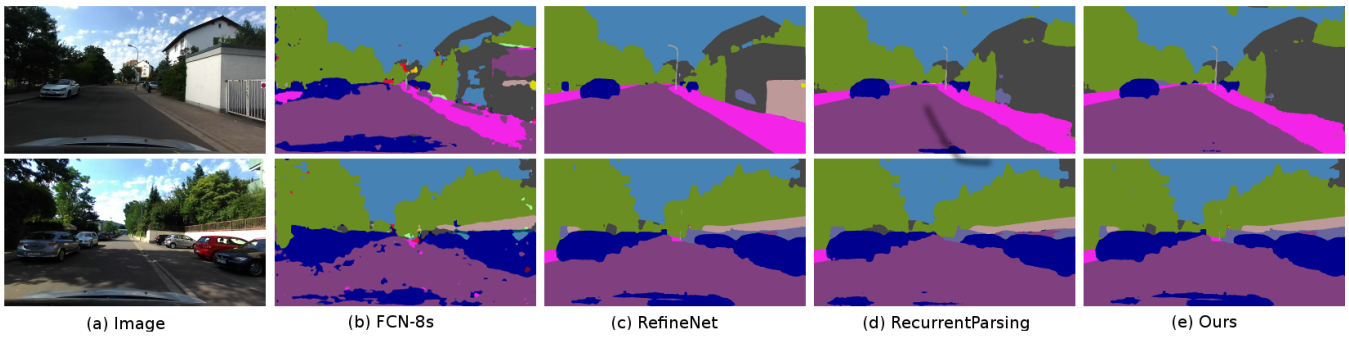

Fig. 4. Qualitative results of our model on self-recorded Zed dataset.

TABLE I

COMPARISON OF THE SUMMATION AND CONCATENATION METHODS FOR THE FUSION BLOCK IN TERMS OF EPOCHS AND MEAN IOU.

	Mean IoU	
Epochs	summation	concatenation
24	73.5	72.4
50	74.1	74.1
60	74.6	74.6
70	74.6	74.7

Batch Normalization + ReLU blocks with different dilation rate. We modified 2^{nd} till 5^{th} pyramid level convolution operations with 3×3 filters with dilation rate $\{2, 4, 8, 16\}$ respectively. The 1^{st} pyramid level convolution operation uses 1×1 filters without dilation rate as in [16]. With small dilation rate the contextual information from the scene is lost as stated in [4], [15], [16]. As compensation, we employ global average pooling [15] which can retain the contextual information making the feature maps more responsive with context to nearby objects (eg. pedestrian near grass or a pedestrian near a car). The input of fusion block is fed to global average pooling which generated the average feature maps. We employ summation to merge the outputs of all pyramid level convolution operations. The outputs from pyramid pooling and global average pooling are concatenated, and the resulting features is passed to 1×1 convolution to produce final *logits* (class-wise probabilistic map along the depth channel of the feature map). We use bilinear interpolation of *logits* to get full resolution for our prediction maps. The combination of dilation rate $\{2, 4, 8, 16\}$, global average pooling along with fusion from the depth branch, we have proven that our results are better when recognizing small objects compared to other methods. This can be verified from the quantitative and qualitative evaluations in the Section IV.

To summarize, our propose model consists of four parts. First part is feature extractors which is composed by two branches of Atrous ResNet-101 to extract features of RGB and depth images. Second part is fusion block that fuses feature maps from both RGB and depth. Third part is global average pooling which is useful to retain the contextual information in the feature maps. The last part consists of pyramid pooling and final prediction. The pyramid pooling is composed of five convolution layers that are arranged in

a parallel order.

D. Implementation Details

We use pre-trained ResNet-101 [29] on ImageNet [31] for feature maps generation in RGB and depth branches separately. The original ResNet-101 requires three channels (RGB) image as an input. Therefore, in our depth branch, we modify the first convolution layer to accept one channel image. We initialize the weights for the first convolution layer in depth branch by averaging the weights of the first convolution layer in RGB branch along the channel dimension. In contrast with the original ResNet-101 [29] we remove the first 7×7 convolution layer and change to three 3×3 convolution layers as in [3], [24]. This modification keeps the receptive field similar to additional parameters to learn and makes the network deeper. We adjust the *output_stride* $= 8$ by adding dilation $rate = 2$ and $rate = 4$ in the last two residual blocks (res4 and res5 in our naming notation) respectively (see Figure 2). We upsample the top-most feature maps with bilinear interpolation by a factor of eight to make the final prediction of full resolution. Our implementation is built on top of the open-source code provided by Kong and Fowlkes [3] which uses MatConvNet [32] framework.

E. Training Protocol

We train our models with the following procedure: Initially, we train RGB and depth branch separately. We insert fusion block before pyramid pooling to combine feature maps generated from both Atrous ResNet-101 from the RGB and depth trained models, which is then passed to pyramid pooling and later for final prediction. Finally, we freeze Atrous ResNet-101 for training Fusion Block and Pyramid Pooling to get our final prediction. This is achieved by setting the learning rate of the freezing layers to zero.

We employ "poly" learning rate policy, in which a base learning rate is multiplied by $(1-\frac{iter}{maxiter})^{momentum}$. We set the base learning rate to 5×10^{-5} and $momentum = 0.9$. Due to limited GPU memory and large image resolution, we set the batch size to one. Atrous convolution requires large cropping size to make dilation rate effective [16]. Therefore, we randomly crop the input images to 720×720 during training on Cityscapes [21]. We set the momentum to 0.9 and weight decay to 0.0005. For data augmentation, we randomly scale the cropped input images with the scale rate between 0.5 and 2.0 and also perform left-right flipping. In the case of training for the RGB model, we also add color jittering. We trained our model for total 200 epochs. From epoch 140 onwards, we change the base learning rate to 5×10^{-4} and weight decay of pyramid pooling layers to 0.999.

IV. EXPERIMENTAL EVALUATION

We evaluate our model on Cityscapes [21]. It is a large-scale outdoor scene dataset recorded across 50 German cities with different seasons. It contains 2975 RGB-D pairs (*train*), 500 (*validation*) and 1525 (*test*) image sets with pixel-wise *fine-annotation* labels, along with $20,000$ extra train RGB-D pairs with *coarse-annotation* labels. We also evaluate on synthetic dataset generated from a publicly available driving simulation framework CARLA [20]. We generated 5000 (*train*) and 500 (*validation*) RGB-D pairs image sets. Furthermore, we perform qualitative evaluation on a self-recorded real data Zed dataset[1]. Zed is captured dataset using a front-facing zed stereo camera [33] mounted on a car.

We provide our quantitative results on the *test* data of Cityscapes and *validation* data of CARLA. We did adjustments in class mapping for CARLA quantitative measurement as follows: First, we ignored two classes, i.e. *road line* and *others*, because they do not exist in Cityscapes. Then we grouped *car, truck* and *bus* classes from Cityscapes to the *vehicles* class in CARLA. Other classes from Cityscapes that do not exist in CARLA are set to *unlabeled* and ignored in mean IoU measurement. We compare our results to other well-known approaches such as FCN-8s [1], RefineNet [2], RecurrentParsing [3], DeepLab-v2(ResNet-101) [4] and PSPNet [15]. We used intersection-over-union (*IoU*) [2] metric for quantitative evaluation. We average the IoU result across 10 classes for CARLA [20] and 19 classes for Cityscapes [21].

In Figure 1, we compare our qualitative results on Cityscapes [21] with the ground truth and other methods. It can be seen that our segmentation is close to ground truth. In comparison to others, we can see that our segmentation stands out of the FCN-8s [1]. Compare to RefineNet [2], we perform better in segmenting narrow *building* and far distant *pole* that can be seen in the 1^{st} row. In Figure 1 4^{th} row, we showed zoomed version of a region from the third-row images. It can be seen that our method segments small size of persons while RefineNet [2] failed to segment them. Moreover, we segment a pole slightly better compare to RecurrentParsing [3].

In Figure 3, we compare our qualitative results on CARLA [20] with the ground truth and other methods. In rows $1^{st} - 3^{rd}$, our approach correctly segments *side walk* where other methods fail. Furthermore, our model correctly segments other objects in the scenes such as *pedestrians, cars, roads, buildings* and *vegetation*. One fail case of our segmentation is in 1^{st} row, where it fails to segment *poles* and *traffic sign* nearby the buildings. It can be because of the same texture and color of the poles with the nearby buildings. Another visual difference from our result with the ground truth is the *sky* segmentation. In all prediction images, the *sky* is labelled with the blue color, while in ground truth it is labelled with black. It happened because the *sky* class does not exist in CARLA [20].

In Figure 4, we perform qualitative evaluation on Zed dataset. Our model is generalized well to real-world data without any fine-tuning or parameters adjustment. We can see that objects such as *buildings, cars, poles* and *vegetations* are segmented correctly even if they are cluttered.

[1]We thank Oliver Wasenmüller for providing the dataset.

[2]$IoU = \frac{TP}{TP+FP+FN}$, where TP is true positive, FP is false positive and FN is false negative pixels.

TABLE II

QUANTITATIVE EVALUATION ON CITYSCAPES DATASET. THE IOU METRIC IS SHOWN IN PERCENTAGE.

Object	FCN-8s	RefineNet	RecurrentParsing	DeepLab-v2(ResNet-101)	Ours
road	97.40	98.20	98.50	97.86	98.45
sidewalk	78.40	83.21	85.44	81.32	85.15
building	89.21	91.28	92.51	90.35	92.24
wall	34.93	47.78	54.41	48.77	47.10
fence	44.23	50.40	60.91	47.36	59.83
pole	47.41	56.11	60.17	49.57	63.12
traffic light	60.08	66.92	72.31	57.86	71.76
traffic sign	65.01	71.30	76.82	67.28	76.79
vegetation	91.41	92.28	93.10	91.85	93.22
terrain	69.29	70.32	71.58	69.43	71.80
sky	93.86	94.75	94.83	94.19	94.62
person	77.13	80.87	85.23	79.83	84.45
rider	51.41	63.28	68.96	59.84	65.66
car	92.62	94.51	95.70	93.71	95.36
truck	35.27	64.56	70.11	56.50	58.11
bus	48.57	76.07	86.54	67.49	73.70
train	46.54	64.27	75.49	57.45	61.99
motorcycle	51.56	62.20	68.30	57.66	66.82
bicycle	66.76	69.95	75.47	68.84	74.13
Mean IoU	65.3	73.6	78.2	70.4	75.4

TABLE III

QUANTITATIVE EVALUATION ON CARLA DATASET. THE IOU METRIC IS SHOWN IN PERCENTAGE.

Object	FCN-8s	RefineNet	RecurrentParsing	PSPNet	Ours
Buildings	57.86	77.05	72.34	71.84	79.08
Fences	14.04	28.53	20.47	25.01	21.78
Pedestrians	9.49	16.60	14.94	18.45	23.94
Poles	16.63	37.28	20.05	27.43	38.27
Roads	79.62	90.69	84.59	81.90	88.96
Sidewalks	26.83	66.14	22.47	10.11	48.45
Vegetation	69.16	70.81	69.99	71.49	68.87
Vehicles	32.73	42.35	58.68	62.93	64.86
Walls	2.49	9.67	4.17	3.43	8.78
Traffic Signs	11.11	26.90	30.21	30.69	27.65
Mean IoU	31.99	46.60	39.79	40.33	47.06

In Table II, we perform quantitative evaluation on Cityscapes [21]. We achieve 75.49% accuracy, which is comparable to other state-of-the-art methods. We gained 5% improvement over the baseline model DeepLab-v2(ResNet-101) [4]. In Table II, our approach achieves better accuracy for small objects such as *pole, traffic sign, person, car, terrain* and *vegetation*, while still having comparable results for big objects. For example in the *pole* class, we gain 13.5% improvement over the baseline DeepLap-v2(ResNet-101) [4] and 2.95% over RecurrentParsing [3].

In Table III, we perform quantitative evaluation on CARLA [20] *validation* data. For a fair comparison, we compare our model with other publicly available Cityscapes-trained models without any fine-tuning or parameters adjustment. Note that our method achieves higher accuracies in several objects such as *buildings, pedestrians, poles* and *vehicles*. It may be worth mentioning that all methods perform poorly in segmenting *walls* where accuracies are less than 10%. One reason for this poor segmentation can be that the different shape and texture between the *walls* in CARLA 3D model and real-world object.

V. CONCLUSIONS

In this paper, we proposed a network to address multi-scale objects in semantic segmentation. A novel combination of depth and multi-scale pyramid network specifically address the multi scale objects segmentation. Our evaluation demonstrated that the proposed network gained 5% improvement over the baseline RGB based method and achieve performance comparable to the state-of-the-art on Cityscapes [21]. Furthermore, we showed that our model is robust to other unknown test sets such as, synthetic images generated from CARLA [20] and on real world images captured using Zed stereo camera [33]. Future work includes training the network with multiple GPUs to accomodate the batch size greater than one for faster training and more robust learning. Additionally, Bayesian learning [34] can increase the overall safety of Autonomous Vehicle (AV) by jointly learning the segmentation and uncertainties.

ACKNOWLEDGMENT

This work was partially funded by the European project *Eyes of Things* under contract number GA643924.

References

[1] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 4, pp. 640–651, 2017.

[2] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," *arXiv preprint arXiv:1611.06612*, 2016.

[3] S. Kong and C. C. Fowlkes, "Recurrent scene parsing with perspective understanding in the loop," *arXiv preprint arXiv:1705.07238*, 2017.

[4] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. PP, no. 99, pp. 1–1, 2017, ISSN: 0162-8828.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.

[6] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[10] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2147–2154.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 6, pp. 1137–1149, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 21–37.

[14] B. Kayalibay, G. Jensen, and P. van der Smagt, "Cnn-based segmentation of medical imaging data," *arXiv preprint arXiv:1701.03056*, 2017.

[15] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[17] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection," in *Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 390–399.

[18] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3640–3649.

[19] I. Kokkinos, "Pushing the boundaries of boundary detection using deep learning," *arXiv preprint arXiv:1511.07386*, 2015.

[20] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[21] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," *arXiv preprint arXiv:1409.1556*, 2014.

[23] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *International Conference on Computer Vision (ICCV)*, 2015, pp. 1520–1528.

[24] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. W. Cottrell, "Understanding convolution for semantic segmentation," *arXiv preprint arXiv:1702.08502*, 2017.

[25] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Asian Conference on Computer Vision (ACCV)*, 2016.

[26] S.-J. Park, K.-S. Hong, and S. Lee, "Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation," in *International Conference on Computer Vision (ICCV)*, 2017.

[27] Y. Cheng, R. Cai, Z. Li, X. Zhao, and K. Huang, "Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR)*, Institute of Electrical and Electronics Engineers, Inc., 2017.

[28] M. D. Ansari, S. Krauß, O. Wasenmüller, and D. Stricker, "Scalenet: Scale invariant network for semantic segmentation in urban driving scenes," Jan. 2018.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[30] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *International Conference on Learning Representations (ICLR)*, 2013.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[32] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*, ACM, 2015, pp. 689–692.

[33] StereoLabs, *ZED Stereo camera*, "`https://www.stereolabs.com/zed/`", Accessed: 2018-01-13.

[34] R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla, and A. V. Weller, "Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning," International Joint Conferences on Artificial Intelligence, Inc., 2017.