Real-time Image Semantic Segmentation Networks with Residual Depth-wise Separable Blocks

Van-Viet Doan [*], Duy-Hung Nguyen [*], Quoc-Long Tran, Do-Van Nguyen, Thanh-Ha Le
UET, Vietnam National University, Hanoi

***Abstract*—Semantic image segmentation plays a key role in obtaining pixel-level understanding of images. In recent years, researchers have tackled this problem by using deep learning methods instead of traditional computer vision methods (eg [25]). Because of the development of technologies like autonomous vehicles and indoor robots, segmentation techniques, that have not only high accuracy but also the capability of running in real-time on embedded platform and mobile devices, are in high demand. In this work, we have proposed a new convolutional module, named Residual depth-wise separable, and a fast and efficient convolutional neural network for segmentation. The proposed method is compared against other state of the art real-time models. The experiment results illustrate that our method is efficient in computation while achieves state of the art performance in term of accuracy.**

***Index Terms*—Image Semantic Segmentation, Residual Learning, Depth-wise Separable Convolution**

I. Introduction

In recent years, there has been a huge development of applications that nourish from information extracted from images such as indoor navigation, augmented reality and autonomous driving. Indoor robots, autonomous cars are machines that are able to automatically navigate and get aware of environment without human instructions. These type of technologies involve in complex computer vision algorithms and variety of sensors and actuators to solve three fundamental problems: navigation and guidance, driving and safety, and performance. The objective of navigation and guidance is to see what is in front, interpret signage and then identify available path, avoid collision with obstacles. To achieve the vision tasks, a camera is used to captured the road scene for autonomous cars and indoor scene for robots. Semantic image segmentation then plays a key role for gaining advanced semantic understanding of these videos input. It provides essential data for followed computer vision algorithm by partition the vehicles, pavements, building and humans into different areas in a frame. Decisions during driving time are made based on the results of image segmentation.

Semantic image segmentation is the process of assigning each pixel a predefined label to simplify the input for analyzation tasks. Deep Convolutional Neural Networks (DCNNs) has significantly boosted the accuracy of semantic segmentation and many others computer vision problems. However, the main problem of this technique is that its model requires large number of parameters and mathematical operations. As new technology devices like autonomous cars and augmented reality glasses are being got interested in by the community, it increases the demand of real-time image semantic segmentation on mobile devices.

In this paper, we develop ~~a~~ CNNs that can operate in real-time on Nvidia Jetson Tx2 (embedded GPU). Our main contribution and statistics are the following:

- We propose a new convolution block that demands twice less computational cost than traditional convolution blocks at the expense of small reduction in accuracy;
- We also proposed a CNNs architecture based on the new block that achieves high quality segmentation with real-time inference time even on low-power mobile devices and embedded system;
- Our approach be able to run at over 21 FPS in a Jetson Tx2 (embedded GPU) with mean intersection over union (mIoU) about 60.2% on public Cityscapes dataset.In other words, the new network provides a good trade-off between inference time and segmentation quality.

II. Related Works

DCNNs was initially introduced for image classification challenges [12]. FCN [14] is the first model that reinforced the use of end-to-end CNNs networks for image semantic segmentation. FCN convolutionalize pretrained network on Imagenet [6] to output feature maps and upsampling the output feature maps. The authors of FCN also proposed skip connections that combine the layers from three pool operations to capture more information because information is lost during multiple pooling time. Other works like Deeplab [4] proposed atrous spatial pyramid pooling to capture multiple scale context or PSPNet [24] introduced pyramid pooling module to inferring different sub-region representations. The work in ICNet [23] based on PSP Net architecture. Instead of feeding high resolution images through network, ICNet re-sizes images with multiple scaling factor then feeding these images through cascade architecture. However, utilizing these technique highly increase computation cost of the network. All these network achieve top accuracy but require high performance GPU and non of them can inference in real-time on embedded devices.

Some other approaches sacrifice accuracy to reduce computational resources such as [3] [16] [10] [1] . Enet [16] introduced a new light weight deep neural network architecture. Encoder and decoder blocks of Enet is bottleneck module it combines building block of resnet and convolution layer 1

This research is granted by American AOARD/AFOSR projects number FA2386-17-1-4053.

[*] Equally contributing authors

x 1 to reduce channels dimension. SQ Net [10] introduced bypass refinement module which memorizes feature maps from encoder layer to improve upsampling. SegNet [1] use unpooling instead of transpose convolution. Unpooling operations require fewer floating point operation when compare to transpose convolution. The work of Linknet [3] make use of convolutional block 1 x 1 to reduce channel dimensions before feeding through transposed convolution, then it uses convolutional block 1 x 1 to regain desired channel dimensions. These method can do real-time semantic segmentation on embedded devices but the accuracy of output semantic map is not sufficient enough for practical applications.

III. Proposed Architecture

We start this section by explaining residual learning [7] and depth-wise separable convolution [9] in details because these two concepts have significant impacts on the principles and basics theories of our proposed method. We actually combined them to form a new type of convolution block. Final part of this section is devoted to the architecture of our model.

A. Residual Leaning

VGG [20] has proved that the network depth is a crucial factor lead to the efficiency of a CNN model. Feature maps are enhanced by stacking more layers. But in practice, when increasing the depth of a suitable network, accuracy gets worse and quickly degrades, this phenomenon is called degradation problem. In idealistic scenario, if we add more layers in a network, the error will be reduced or at least remain the same. This hypothesis can be explained as following: layers that belong to the original network persisted and the added layers just act as an identity mapping. In practice, finding the identity mapping in a form of a stack of nonlinear layers is extremely hard. As Resnet [7] came out in 2015, it solved the degradation problem by introducing residual learning method. Fig. 1 and Fig. 2(b) illustrate residual leaning blocks. The residual function can be defined as:

$$y = M(x) + x \tag{1}$$

where x is the input feature maps and M is the mapping function using stacked non-linear layers. The standard convolution function is y = M(x). In order to satisfy the identity mapping y = x, we need to solve the equation M(x) = 0 for residual function and M(x) = x for standard one. The first mentioned one is easier when we consider that M stand for some stacked non-linear layers. Paper about semantic segmentation recently used widely residual learning as default block for encoder and decoder because this method strongly decreases the effect of degradation while do not involve extra parameters.

B. Depth-wise separable convolution

Depth-wise separable convolution is introduced in MobileNet [9] as a type of factorized convolution. It consists of two layers: depth-wise convolution and point-wise convolution. For example, we have a convolution layer of size $D_R \times D_R \times N_I \times N_O$, feature map of size $D_F \times D_F \times N_I$ as input and a $D_H \times D_H \times N_O$ output feature map where:

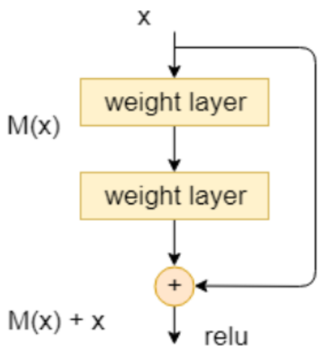

Fig. 1. Residual block

- D_R : the receptive filed (or filter) size;
- D_F : the spatial size of input feature maps;
- D_H : the spatial size of output feature maps;
- N_I : number of input feature maps depth dimension;
- N_O : number of output feature maps depth dimension.

The complexity of standard convolution is:

$$D_R \times D_R \times N_I \times N_O \times D_F \times D_F \tag{2}$$

In depth-wise convolution, each filter is processed on a single input channel. In other words, it only extract information from each channel of feature maps and does not combine them like standard convolution. Therefore, depth-wise convolutions computation cost gets rid of N_O and is defined as:

$$D_R \times D_R \times N_I \times D_F \times D_F \tag{3}$$

Point-wise convolution is similar to standard convolution except that the filter size is 1 x 1. The complexity of point-wise convolution is:

$$N_I \times N_O \times D_F \times D_F \tag{4}$$

Depth-wise separable convolution block is a depth-wise convolution layer followed by a point-wise convolution one. As depth-wise convolution is lack of combining factor, point-wise layers take feature maps from depth-wise as input and combine the information in multi-channels. Depth-wise separable convolution represent convolution operation as two-step process of extracting and compressing to have a complexity of:

$$D_R \times D_R \times N_I \times D_F \times D_F + N_I \times N_O \times D_F \times D_F \tag{5}$$

Compare to standard convolution, the new technique gets a reduction of:

$$\frac{D_R \times D_R \times N_I \times D_F \times D_F + N_I \times N_O \times D_F \times D_F}{D_R \times D_R \times N_I \times N_O \times D_F \times D_F} = \frac{1}{N_O} + \frac{1}{D_K^2} \tag{6}$$

The architecture of depth-wise separable convolution is shown in Fig.2 (b).

C. Residual depth-wise separable blocks

We propose a new block which is combined of depth-wise separable convolution and residual learning. It inherits the advantage of these two technique: overcome degradation problem and greatly reduce computational cost. We call it residual depth-wise separable (RDS) blocks, it is illustrated in Fig 2 (c). Similar to depth-wise separable block, each layer is followed by batch normalizations as it reduces the convergence time. We replace point-wise convolution with a 3x3 convolution in order to gather more context representation and expand the field of view. One drawback of depth-wise separable convolution is that each channel is only related to a certain channels and lack of cross talk to other ones like standard convolution. Therefore, we adopt the idea of channel shuffle operation [22] to boost the accuracy. Shuffle channels with the group numbers (a hyper-parameter of channel shuffle operation) of 4 are added after 3x3 convolution.

RDS block have the computational cost of:

$$3 \times 3 \times N_I \times D_F \times D_F + 3 \times 3 \times N_I \times N_I \times D_F \times D_F \quad (7)$$

Residual block has the computational cost of:

$$2 \times 3 \times 3 \times N_I \times N_I \times D_F \times D_F \quad (8)$$

Compare to residual block, RDS get a reduction in complexity of:

$$\frac{3 \times 3 \times N_I \times D_F \times D_F + 3 \times 3 \times N_I \times N_I \times D_F \times D_F}{2 \times 3 \times 3 \times N_I \times N_I \times D_F \times D_F} = \frac{1}{2 \times N_I} + \frac{1}{2} \quad (9)$$

Because N_I is greater than 2^6 in our model, the left part can be ignored. Express in other words, RDS block is two times faster than residual block. The efficiency of the new block will be demonstrated in the experiments and results section.

D. Architecture design

The architecture design uses our proposed RDS block in many layers. To get an acceptable trade-off between the performance and inference time, the model follows some strategies:

- **Early down-sample**: We adopt the view of [16] that using the first two layer to downsize the image by 4. It has two benefits : significantly reduce the process time and infer features from early stage to let the following layers gain more context and enlarge view size;
- **Asymmetric encoder-decoder architecture**: Decoder is just bi-linear interpolation layer and all of parameterized layers concentrates in encoder portion. Because the encoder is the most crucial part as it detects feature, extracts valuable information whereas the main target of decoder is only up-sampling feature maps from encoder. Besides, using only a bi-linear layer as decoder also speeds up the inference time of our model.
- **Large feature maps resolution**: Unlike SegNet and FCN that downsize the image 5 times, we only down-sample by 3 times. Heavily down-sampling involve heavily up-sampling, it leads to increasing the inference time. Besides, low resolution maps lose details and edge information that are captured by low-level filters. Apply this strategy, our model even effectively predicts small objects such as poles and sign symbols.

The proposed architecture is illustrated in Table I. For down-sample block, we use max-pooling and convolution 3x3 with stride 2 in parallel and then do concatenation like the one that is described in [16]. We add dropouts [8] in all encoder RDS blocks as it avoids over-fitting and improve the accuracy. We also use dilated convolution [4] to capture more context representation while remain the same computation cost and memory space. A 3x3 convolution with stride 1 is attached to encoder, the main task of it is turning the feature map into a low-resolution segmentation images. Then decoder do the task of up-sampling these images into the original size.

IV. Experiments

A. Datasets

Cityscapes [5] is a high quality dataset with varying weather condition, daytimes and road scenarios. The original data resolution is 2048 x 1024 which were captured in 50 cities and focused on urban street scenes. It consists of 5000 fine-annotated images: 2975 for training, 500 for validation and 1525 images are used for testing. The dataset also delivers dense pixel annotations for 30 classes grouped into 8 categories including sky,nature, objects, construction, humans, flat surfaces, vehicles, and void. It has the large number of scene layouts, highly varying background and dynamic objects

Camvid [2] is a road scene understanding database which consist of 700 fine-annotated images: 367 images are used for training, 100 for validation and 233 for testing. It provides 12 label class including sky, building, pole, road, pavement, tree, sign symbol, fence, car, pedestrian, bicyclist and unlabeled object. It was original captured as video and the frames are then sampled at 1 fps and 15 fps.

SUNCG [21] is a synthetic indoor image dataset consists of more than 45,000 scenes, each has different rooms such as living room, bed room, kitchen, etc. Each room has different type of 91 objects including chair, sofa, computer, freezer ... Those environments are for those who want to test simulated robot perception/navigation algorithms in simulation such as MINOS [19] before applying in real. For the experiment, we collect data from 51 scenes consist of 14,000 pairs of RGB/segment images in the semantic segmentation task.

B. Experiment setting

We used PyTorch framework [17] to implement CNN network as well as train, test and evaluate the model. It took one day to train the proposed model on Nvidia Tesla K40m GPU.

Most of the state the art methods use a pretrained CNN model on ImageNet or COCO [13] as encoder and then fine tune the new network on target datasets. In other words, these models were trained on 2 different datasets and the accuracy is significantly boosted by over 3 million images on ImageNet

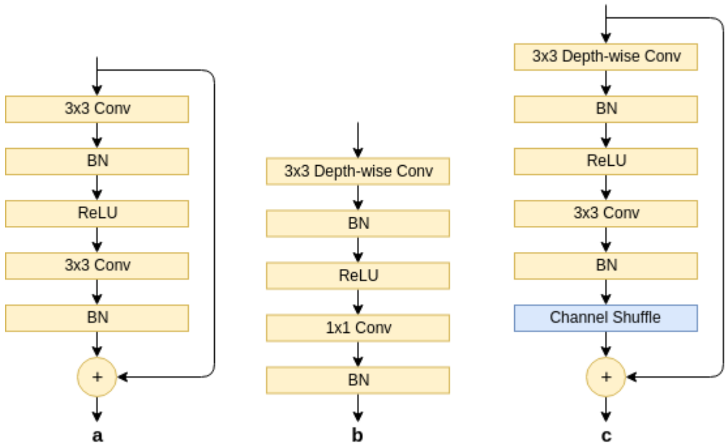

Fig. 2. a) Rediual block. b) Depth-wise separable block. c) RDS block .

TABLE I
OUR PROPOSED ARCHITECTURE DESIGN.

	Type	Output size	Dilation rate	Dropout
Encoder	Downsample Block	32 x 512 x 256	None	None
	Downsample Block	64 x 256 x 128	None	None
	RDS block	64 x 256 x 128	None	0.5
	RDS block	64 x 256 x 128	None	0.5
	RDS block	64 x 256 x 128	None	0.5
	Downsample Block	128 x 128 x 64	None	None
	RDS block	128 x 128 x 64	None	0.3
	RDS block	128 x 128 x 64	2	0.3
	RDS block	128 x 128 x 64	4	0.3
	RDS block	128 x 128 x 64	8	0.3
	RDS block	128 x 128 x 64	16	0.3
	3 x 3 Convolution	C x 128 x 64	None	None
Decoder	Bilinear interpolation	C x 1024 x 512	None	None

dataset. Because of the limitation of computational hardware, training our encoder network on ImageNet is unreachable. Therefore, we decided to just train the whole network on the target datasets.

For training, we use Adam [11] optimization as it is the most effective algorithm and currently recommended as the default algorithm to use. We train the network with learning rate of 5e-4 and weight decay of 2e-4, and divide the learning rate by 2 when the loss is stagnant. The data is augmented by random translation of 0-2 pixels on both axes and horizontal flips.

We use cross entropy 2D as loss function. It can be described as:

$$loss(p, cl) = w[cl](-p[cl] + log(\sum_j e^{p[j]})) \tag{10}$$

Where p is scores for each class of pixel, cl is the number of classes and w is the vector assigning weight to each of the classes. It is essential to add weight argument as the training dataset is unbalanced. For example, in road scene image, the majority of pixels belongs to road and building while the number of poles pixels is extremely small. The model then tends to predict a pixel belongs to a common class. The weight is taken from [16] and described as:

$$w[i] = \frac{1}{log(c + n_i)} \tag{11}$$

Where c is a hyper parameter that is used to limit the weight values between a predefined interval, n_i is the appearance frequency of pixels belong to class i.

C. Results

Table II , Table III show the accuracy results for Cityscapes and Camvid dataset, respectively. The reader might find some comparison on predicted images at Fig. 4 in appendix section. All reported accuracy results are obtained from associated papers. We compare our accuracy with other real-time network.

TABLE II
CITYSCAPES TEST SET RESULTS.

Model	Mean IoU
SegNet	57.0
Enet	58.3
SQNet	59.8
ESPNet	60.2
RDS-Net(Ours)	60.2
ICNet	69.5
ErfNet	**69.8**

TABLE III
CAMVID TEST SET RESULTS.

model	Mean IoU
Enet	51.3
SegNet	55.6
ESPNet	55.6
RDS-Net(Ours)	**58.3**

Our model achieves the highest accuracy, at 58.3%, on Camvid and is ranked third on Cityscapes with 60.2%. Although the two highest one on Cityscapes, ICNet and ErfNet [18], is considered as a real-time architectures, its time benchmark actually is evaluated on Titan X, a powerful GPU. In practice, it is not fast enough to be used in embedded system and low-power devices.

We also retrain and test some networks. For ESPNet [15] and ErfNet, we use the public code of the authors to evaluate. For Enet, we ourselves implement this model on pytorch. For SUNCG dataset Fig. 3 notices that our model converges at a better place to compare with the rest. Table IV also shows that our network stand on the top accuracy measurements except for one that is slightly less than ERF-Net. Fig 5 in appendix section is for further reference on prediction results.

Table V compare inference time of varying resolution images on embedded GPU NVIDIA Jetson TX2 and a laptop GPU GTX 1050. Our model is as fast as the fastest one, ESPNet, while is 2.7% more accurate on camvid and has the same accuracy in Cityscapes. Our model can do inference 20 fps for image of resolution 640 x 360 which is suitable for real-time applications.

V. Conclusion

In this paper, we have proposed a real time semantic segmentation network with Residual Depth-wise Separable blocks which could be mounted to mobile devices and embedded systems. The new network design is motivated by effective convolutional blocks that overcome degradation problem and speed up the inference time. We also consider some of the important design strategies to improve the performance. Our experiments prove that the proposed network delivers a good trade-off between reliability and speed.

Because of the constraints of computational system, we did not use all external dataset in our experiment even though it can significantly increase the accuracy. But it turns out that our model is robust enough to be trained from scratch in the target dataset thereby simplify experiment and reduce training time. Future related researches need to focus on creating new type of factorized blocks that replace the old standard ones for real-time segmentation problems, and build the new light-weight CNNs architecture from scratch instead of depending on transfer learning methods that is too slow.

Acknowledgment

This research is granted by American AOARD/AFOSR projects number FA2386-17-1-4053. We thank to Hai-Dang Nguyen and Tai-Long Nguyen for their contribution in data preparation and further experiments.

References

[1] Badrinarayanan, Vijay, Kendall, Alex, and Cipolla, Roberto. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[2] Brostow, Gabriel J et al. “Segmentation and recognition using structure from motion point clouds”. In: *European conference on computer vision*. Springer. 2008, pp. 44–57.

[3] Chaurasia, Abhishek and Culurciello, Eugenio. “LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation”. In: *arXiv preprint arXiv:1707.03718* (2017).

[4] Chen, Liang-Chieh et al. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2018), pp. 834–848.

[5] Cordts, Marius et al. “The cityscapes dataset for semantic urban scene understanding”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 3213–3223.

[6] Deng, Jia et al. “Imagenet: A large-scale hierarchical image database”. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE. 2009, pp. 248–255.

[7] He, Kaiming et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[8] Hinton, Geoffrey E et al. “Improving neural networks by preventing co-adaptation of feature detectors”. In: *arXiv preprint arXiv:1207.0580* (2012).

[9] Howard, Andrew G et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

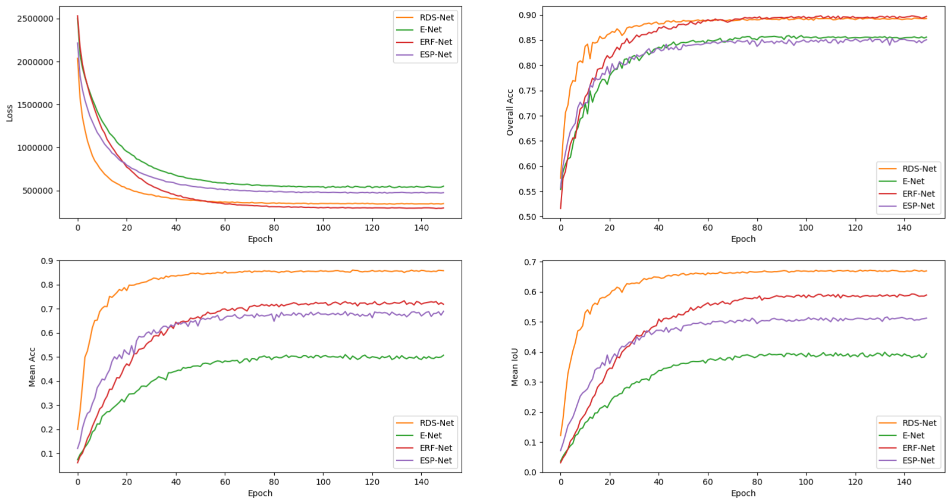

Fig. 3. Training plot

TABLE IV
SUNCG TEST SET RESULTS.

Model	Overall Acc	Mean Acc	FreqW Acc	Mean IoU
Enet	84.2	55.6	74.8	42.7
ERF-Net	**92.6**	73.8	**86.8**	63.8
ESPNet	87.9	67.8	79.7	55.6
RDS-Net(Ours)	**92.0**	**86.8**	85.7	**72.5**

TABLE V
PERFORMANCE OF FASTEST NETWORK ON GTX 1050 AND NVIDIA JETSON TX1

Model	NVIDIA Jetson Tx1						GTX 1050					
	480 x 320		640 x 360		1280 x 720		480 x 320		640 x 360		1280 x 720	
	ms	fps	ms	fps	ms	fps	ms	fps	ms	fps	ms	fps
Ours	41	24.4	49	20.4	188	5.3	**9**	**111**	**13**	**77.0**	**48**	**20.8**
ESPNet	**35**	**28.6**	**46**	**21.7**	**180**	**5.6**	**9**	**111**	14	71.4	49	20.4
Enet	55	18.2	65	15.4	253	4.0	15	66.7	20	50	73	13.7
Erfnet	99	10.1	127	7.9	484	2.1	29	34.5	37	27.0	141	7.1

[10] Iandola, Forrest N et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size”. In: *arXiv preprint arXiv:1602.07360* (2016).

[11] Kingma, Diederik P and Ba, Jimmy. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[12] Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[13] Lin, Tsung-Yi et al. “Microsoft coco: Common objects in context”. In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[14] Long, Jonathan, Shelhamer, Evan, and Darrell, Trevor. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[15] Mehta, Sachin et al. “ESPNet: Efficient Spatial Pyramid of Dilated Convolutions for Semantic Segmentation”. In: *arXiv preprint arXiv:1803.06815* (2018).

[16] Paszke, Adam et al. “Enet: A deep neural network architecture for real-time semantic segmentation”. In: *arXiv preprint arXiv:1606.02147* (2016).

[17] Paszke, Adam et al. *Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration, may 2017*.

[18] Romera, Eduardo et al. “ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.1 (2018), pp. 263–272.

[19] Savva, Manolis et al. “MINOS: Multimodal Indoor Simulator for Navigation in Complex Environments”. In: *arXiv:1712.03931* (2017).

[20] Simonyan, Karen and Zisserman, Andrew. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[21] Song, Shuran et al. “Semantic Scene Completion from a Single Depth Image”. In: *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition* (2017).

[22] Zhang, Xiangyu et al. “Shufflenet: An extremely efficient convolutional neural network for mobile devices”. In: *arXiv preprint arXiv:1707.01083* (2017).

[23] Zhao, Hengshuang et al. “Icnet for real-time semantic segmentation on high-resolution images”. In: *arXiv preprint arXiv:1704.08545* (2017).

[24] Zhao, Hengshuang et al. “Pyramid scene parsing network”. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2881–2890.

[25] Zheng, Shuai et al. “Conditional random fields as recurrent neural networks”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1529–1537.

Appendix

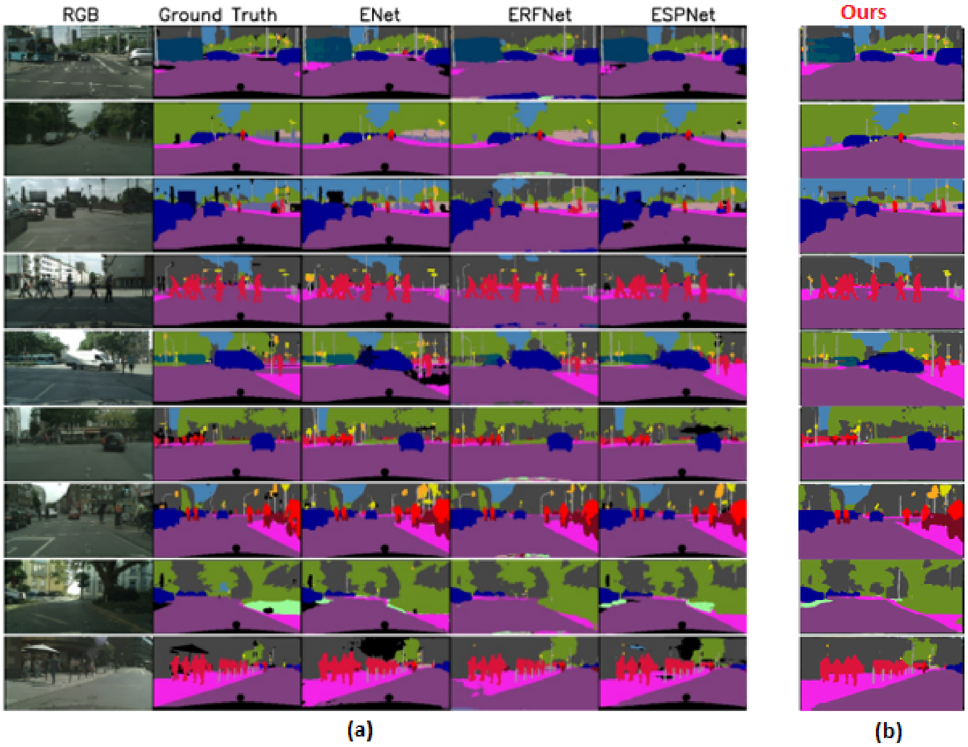

Fig. 4. Comparing on Cityscapes validation dataset (a) results from [15] (b) RDS-Net(Ours)

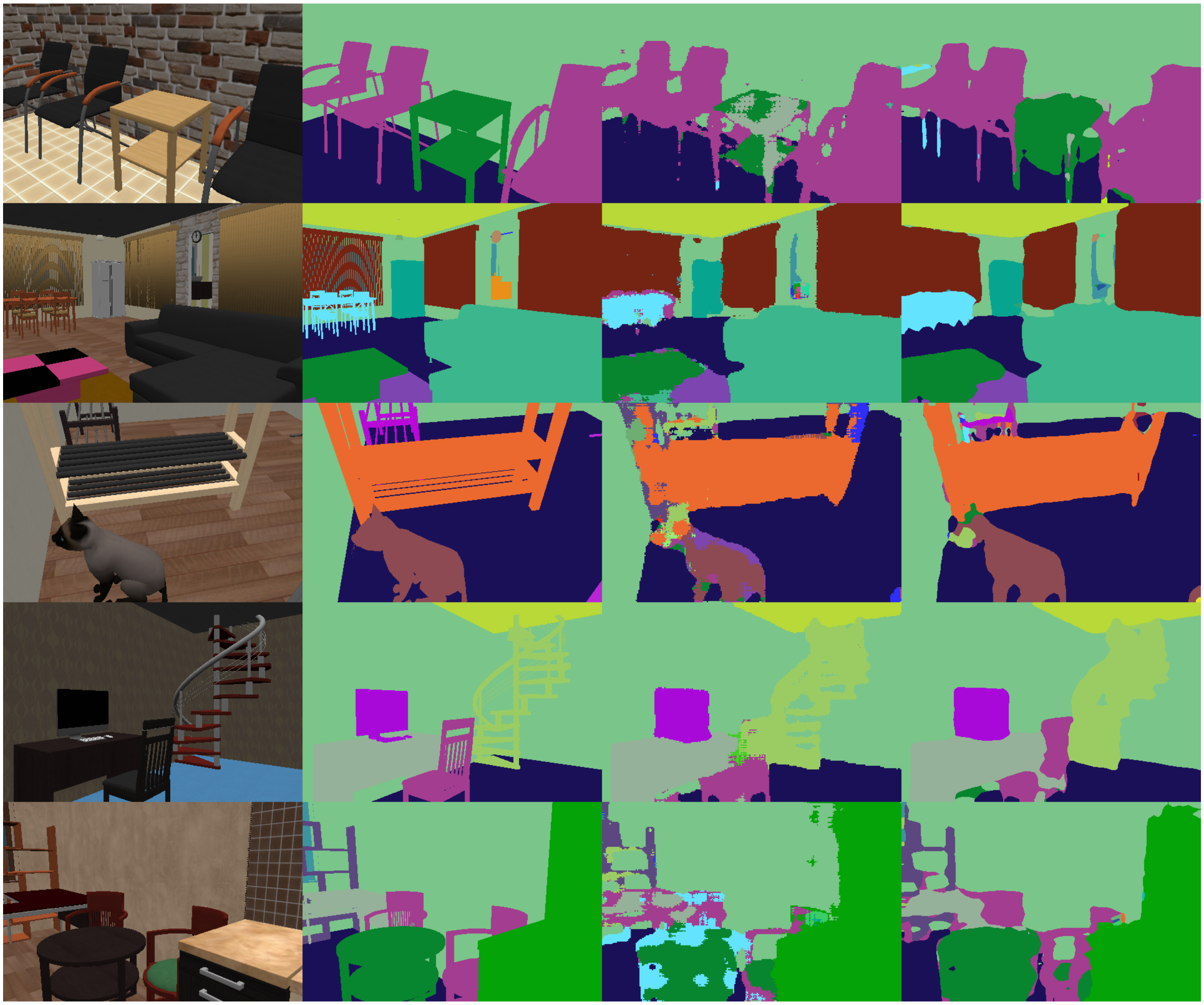

Fig. 5. Comparing on SUNCG Dataset (left to right): RGB, Ground truth, Erf-Net, RDS-Net(Ours)