DEEP SEMANTIC SEGMENTATION FOR THE OFF-ROAD AUTONOMOUS DRIVING

I. Sgibnev *, A. Sorokin , B. Vishnyakov, Y. Vizilter

FGUP «State Research Institute of Aviation Systems», Russia, 125319, Moscow, Viktorenko street, 7 - (sgibnev, ans, vishnyakov, viz)@gosniias.ru

KEY WORDS: Semantic segmentation, DCNN, off-road, autonomous driving, lightweight architectures

ABSTRACT:
This paper is devoted to the problem of image semantic segmentation for machine vision system of off-road autonomous robotic vehicle. Most modern convolutional neural networks require large computing resources that go beyond the capabilities of many robotic platforms. Therefore, the main drawback of such models is extremely high complexity of the convolutional neural network used, whereas tasks in real applications must be performed on devices with limited resources in real-time. This paper focuses on the practical application of modern lightweight architectures as applied to the task of semantic segmentation on mobile robotic systems. The article discusses backbones based on ResNet18, ResNet34, MobileNetV2, ShuffleNetV2, EfficientNet-B0 and decoders based on U-Net and DeepLabV3 as well as additional components that can increase the accuracy of segmentation and reduce the inference time. In this paper we propose a model using ResNet34 and DeepLabV3 decoding with Squeeze & Excitation blocks that was optimal in terms of inference time and accuracy. We also demonstrate our off-road dataset and simulated dataset for semantic segmentation. Furthermore, we present that using pre-trained weights on simulated dataset achieves to increase 2.7% mIoU on our off-road dataset compared pre-trained weights on the Cityscapes. Moreover, we achieve 75.6% mIoU on the Cityscapes validation set and 85.2% mIoU on our off-road validation set with a speed of 37 FPS for a 1,024×1,024 input on one NVIDIA GeForce RTX 2080 card using NVIDIA TensorRT.

1. INTRODUCTION

Reliable and stable semantic model of the surrounding scene, detection of objects and all kinds of obstacles that may appear in the path of an autonomous car is a difficult task for any machine vision system.

Object detection is a two-step approach. At first, we need to localize the instances of interest in the image, then to classify them. Using deep convolutional neural networks, we can build a bounding box for each object in the image. However, this approach does not convey the exact shape of the object and does not consider the entire context of the image because the bounding boxes are rectangular. Therefore, object detection does not provide a complete understanding of the surrounding scene.

Semantic segmentation is essentially a pixel-by-pixel classification, so it gives a more detailed view of the shape of objects in an image and provides a much more complete understanding of the surrounding scene compared to the detection methods. Today we can see an increasing number of applications of semantic segmentation, such as autonomous vehicles, robotic systems and virtual reality for which an understanding of the scene is necessary. Image semantic segmentation is crucially important for the automatic control system of modern autonomous vehicles. An accurate understanding of the surrounding scene is important for navigation and decision-making by control system of robotic platform.

A vision system based on semantic segmentation algorithms is one of the key elements of an off-road autonomous robotic vehicle. Its characteristics largely determine the efficiency of the robotic complex, as it directly affects such problems as recognition of the underlying surface type, calculation of patency map, accuracy of detection, recognition and tracking of objects and obstacles. The imposition of semantic segmentation on a three-dimensional model or point cloud gives us the class of each point and adjust the patency map of the robotic vehicle.
Currently, the task of semantic segmentation is being generally solved by using convolutional neural networks, which can take an image of arbitrary size as an input and output an appropriate predict. New methods that are based on deep convolution neural networks significantly outperform old methods, based on clustering, histogram and color, compression, edge detection, etc.

2. RESEARCH OVERVIEW

2.1 Lightweight backbones

In (Kaiming He et al., 2015) there was presented ResNet, which was able to solve the problem of a vanishing gradient in the process of training deep neural networks by adding shortcut connections. Scientists were given a way to train deeper neural networks than was previously possible. The authors in numerous experiments demonstrated the possibility of effective training of deep neural networks. The results obtained at various competitions made ResNet one of the most popular architectures for solving various problems of computer vision. MobileNetV2 (Mark Sandler et al., 2018) was designed specifically for mobile devices. The authors sought to create a model that would provide high accuracy with a minimum number of parameters and FLOPs. It was necessary to apply this model to solve various computer vision tasks on devices with limited resources. MobileNetV2 bottleneck with expansion layer block is based on depthwise and pointwise convolutions, which allowed authors to significantly reduce the number of parameters and calculations. In ShuffleNetV2 (Ningning Ma et al., 2018) there were added pointwise group convolution and channel shuffling used to exchange information between channels of feature maps. This neural network focuses on maintaining maximum accuracy with significant computational limitations (<200 MFLOPS), thereby focusing on applications for mobile phones, robots, drones, etc.

In (Mingxing Tan et al., 2019) the authors created basic neural

* Corresponding author

Figure 1. Sample images from our off-road datasets

network by doing a Neural Architecture Search then scaled it along different dimensions using proposed scaling. Thus, there were presented several models with balancing network width, depth, and resolution to any resource constraints with maintaining model efficiency – one of such CNNs was EfficientNet-B0.

Models that are based on such neural networks allow us to solve the problem of semantic segmentation on devices with limited resources.

2.2 Decoders

Modern models that solve the problem of semantic segmentation are mostly based on encoder-decoder networks, which are also successfully used to solve many computer vision tasks such as object detection, pose estimation, etc. However, due to the presence of objects of various shapes and sizes on the image, they have a problem with the classification of small objects. To solve this problem, the architecture of the neural network DeepLabV3 was presented in (Liang Chieh Chen et al., 2017). As a decoder, Atrous Spatial Pyramid Pooling (ASPP) was developed to effectively increase the receptive field of feature maps by using dilation convolution with different dilation rate. Therefore, ASPP provides quality descriptors for objects of various sizes.

U-Net (Olaf Ronneberger, et al., 2015) was developed especially for Biomedical Image Segmentation and contained two paths such us encoder and decoder. U-Net is an improved version of the simple SegNet (Vijay Badrinarayanan, et al., 2015) in which authors added skip connections to decoder to use encoder feature maps with upper levels of the convolutional neural network to increase accuracy.

Experiments confirm the effectiveness of these decoders in semantic segmentation tasks

2.3 Additional components

In work (Jie Hu et al., 2017) Squeeze & Excitation block was presented, that could be integrated into the architecture of any convolutional neural network. Using this module, recalibration of feature maps is carried out, which increases the components of the strong features and reduces the components of the weak ones. Moreover, a slight increase in the complexity of the model is accompanied by a significant increase in the accuracy of segmentation.

Image augmentation is a widely used technique for increasing the size and variety of datasets. Deep learning frameworks implement basic image transformations such as reflection, scaling, rotation, etc. Albumentations (A. Buslaev et al., 2018) is an efficient data augmentation package which designed specifically for image augmentation. This tool provides many additional transformations, such as RGBShift, ChannelShuffle, Blur, etc., which can be composed in complex pipelines.

3. PROPOSED METHOD

3.1 Off-road datasets

Recently, semantic segmentation algorithms have been actively developed due to their application in various fields. Autonomous transport is one of the ways to apply these algorithms.

However, most databases of images, annotated with segmentation masks, were collected in urban street scenes. This implies the presence of buildings, paved roads, sidewalks, pedestrians and many different vehicles. Therefore, for semantic off-road scene understanding we created our original dataset consisting of around 100,000 annotated images, in which we included forests, groups of trees, bushes, embankments, ravines, ditches, stones, fields, various types of dirt roads, buildings, structures and other types of obstacles. It was captured in the countryside at every time of the year, at different times of day, in different weather conditions.

Subsequently, we found the terrain and lighting conditions in which the models predicted wrong labels, so we added more data for them. We defined 14 classes: hard ground, soft ground, building, fence, impassable vegetation, passable vegetation, sky, people, vehicle, water, traffic sign, pole, other obstacles and void.

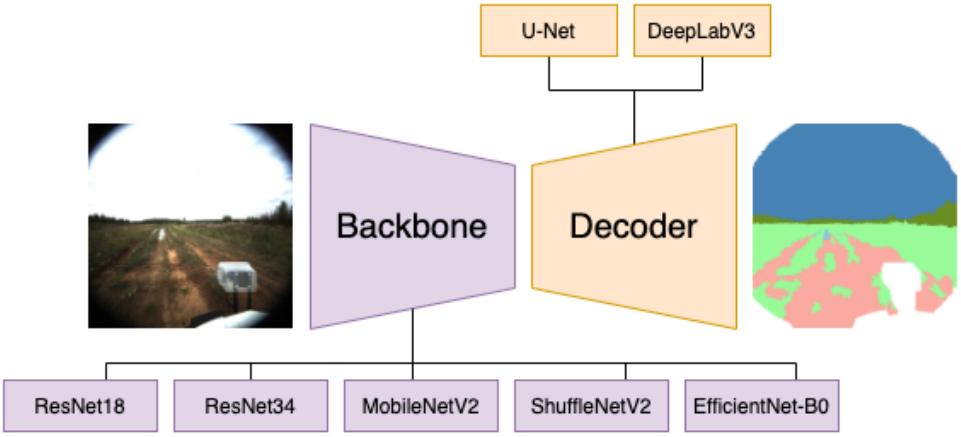

Figure 2. Proposed backbones and decoders

Moreover, we created simulated off-road dataset consisting of 3,500,000 images, around 10% of them being automatically annotated with segmentation masks. We used our own software package based on Unreal Engine 4 graphics engine that provides a large set of tools for creating 3D projects. “Forest terrain”, “hilly terrain”, “wetland terrain”, “snowy terrain” and “mountain landscape” scenes with different weather conditions and lighting settings were modelled. The resulting maps have a unique landscape covering an area of about 4 to 20 square kilometers. On each map, more than 120 routes were created by which we virtually drove more than 200 times having various objects and obstacles in the field of view of our virtual cameras. In this dataset we included objects of 7 classes: ground, building, impassable vegetation, passable vegetation, sky, water and other obstacles.

3.2 Meta-Architecture

Encapsulating lightweight models as a backbone architecture in decoders can achieve significant performance boost with a relatively little loss in accuracy. We propose an architecture (Figure 2) that can handle U-Net and DeepLabV3 as a decoder, inspired by works (Mennatullah Siam, et al., 2018), (Mostafa Gamal, et al., 2018), (Vladimir Iglovikov, et al., 2018) and (Pavel Yakubovskiy, 2020).

We experimented with Resnet18, Resnet34, MobileNetV2, ShuffleNetV2 and EfficientNet-B0 backbones that demonstrated similar results on the ImageNet dataset. Also, we used various pre-trained weights for model training to compare them and achieve boost of accuracy of semantic segmentation. In Table 1 we show Top-1 error percent of our backbones, achieved on ImageNet dataset.

Method	Top-1 error (%)
ResNet18	30.2
ResNet34	26.7
MobileNetV2	28.1
ShuffleNetV2	30.6
EfficientNet-B0	23.7

Table 1. Top-1 error on ImageNet

We transformed encoders and decoders, trained using PyTorch, into ONNX format that is supported by TensorRT.

Datasets were divided into train set (80%) and validation set (20%). We evaluate our models on simulated dataset, original off-road dataset and Cityscapes (Cordts et.al, 2016). Intersection-over-union (IoU) metric is used as an assessment of accuracy.

4. EXPERIMENTAL RESULTS

4.1 Training

Firstly, we trained models on our simulated dataset that was created on the Unreal Engine 4 and on the Cityscapes train set to get pre-trained weights.

Decoder	Backbone	mIoU (%)
U-Net	ResNet18	68.8
	ResNet34	70.3
	MobileNetV2	67.5
	ShuffleNetV2	67.0
	EfficientNet-B0	65.3
DeeplabV3	ResNet18	95.2
	ResNet34	**95.5**
	MobileNetV2	94.5
	ShuffleNetV2	93.2
	EfficientNet-B0	90.1

Table 2. Our simulated validation set results

Pretrained	Decoder	Backbone	mIoU (%)
Cityscapes	U-Net	ResNet18	62.4
		ResNet34	63.9
		MobileNetV2	60.7
		ShuffleNetV2	57.3
		EfficientNet-B0	54.1
	DeeplabV3	ResNet18	80.8
		ResNet34	82.5
		MobileNetV2	78.1
		ShuffleNetV2	75.3
		EfficientNet-B0	72.8
Our simulated dataset	U-Net	ResNet18	64.0
		ResNet34	66.4
		MobileNetV2	63.5
		ShuffleNetV2	59.1
		EfficientNet-B0	57.1
	DeeplabV3	ResNet18	83.3
		ResNet34	**85.2**
		MobileNetV2	81.0
		ShuffleNetV2	78.7
		EfficientNet-B0	74.9

Table 3. Our off-road validation set results

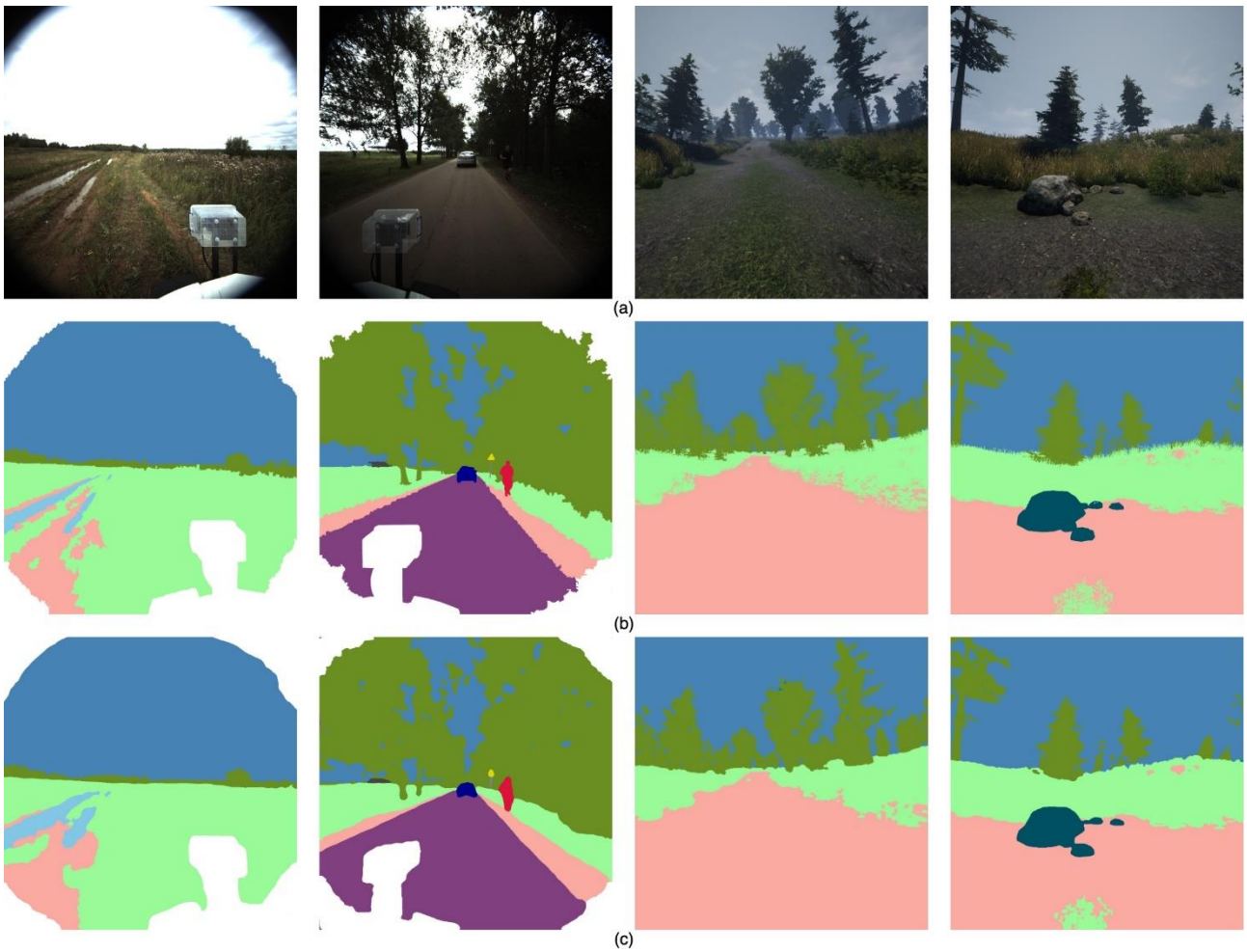

Figure 3. Input images from our off-road datasets (a), ground-truth images for our datasets (b), result images of the proposed approach (c)

Secondary, we used these pre-trained weights to finetune models on our original off-road dataset to compare results. Using the pre-trained weights on our simulated dataset, model with backbone of ResNet34 and DeeplabV3 decoding increased 2.7% mIoU compared to the pre-trained weights on the Cityscapes dataset (Table 3).

This increase in accuracy allows us to use ResNet34 as encoder instead of more heavyweight backbone such as ResNet50, ResNet101, etc. At the same time, we improve inference time and get a comparable accuracy that is a distinct advantage of this approach.

Thirdly, using pre-trained weights on our original dataset we finetuned models on the Cityscapes training set and tested on the Cityscapes validation set.

Method	mIoU (%)	Time(ms)
HRNetV2	81.1	-
DeepLabv3 (ResNet101+ASPP)	78.5	491
Our (ResNet34 + DeeplabV3)	**75.6**	**157**

Table 4. Cityscapes validation set results for 2,048×1,024 input on PyTorch

In Table 4 we show results on Cityscapes validation set, calculated on NVIDIA GeForce RTX 2080 Ti in PyTorch. Method with extractor of ResNet34 and DeepLabV3 decoder with SE blocks demonstrated optimal result in terms of inference time and accuracy on all tests.

To increase the diversity of the train set, a standard set of techniques was used: flipping, cropping, rotating, scaling and their compose. Also, we used «Albumentations» (A. Buslaev et al., 2018) augmentation such as «Cutout», «Hue Saturation Value», «Random Brightness Contrast», «Random Gamma», «RGB Shift» to improve accuracy.

At the pretraining stage, we used backbone, pre-trained on the ImageNet dataset, Adam optimizer, batch size = 8, learning rate = 0.01, which changes by cyclical learning rate politics.

Final models were trained with stochastic gradient descent. As a result of the experiments, optimal parameters were established: batch size = 8, learning rate = 0.001, which decreased every epoch using reduce learning rate on plateau politics, momentum = 0.90 and Nesterov momentum update. The models and learning steps were implemented in PyTorch 1.2.

4.1 Inference

The inference of neural networks is expected to have minimal latency, maximum throughput, optimal memory consumption usage and power efficiency.

Inference of model can be implemented using deep learning frameworks (Caffe, MxNet, Keras, Tensorflow, PyTorch) and special compiler-optimizers that rebuild the neural network architecture for a hardware device (CPU, GPU, NPU). PyTorch is an extremely useful tool for training neural networks, but it does not provide benefits in inference time on GPU. Therefore, we used compiler-optimizer NVIDIA TensorRT which performs optimization of a neural network for NVIDIA GPU platforms. This tool allows to speed up the inference time using various optimizations such as vertical and horizontal layer fusion, etc.

TensorRT optimizes the network by combining layers and optimizing kernel selection for improved latency, throughput, power efficiency, and memory consumption. We created our own high-level library to perform all image processing operations (resizing, transposition, channel swap, etc.) on GPU.

NVIDIA TensorRT takes a model of a neural network that has been converted from PyTorch to ONNX as an input parameter, and serializes engine.

Method	Time(ms) on PyTorch (NVIDIA 2080 Ti)	Time(ms) on TensorRT (NVIDIA 2080, fp16)
ResNet18 + DeepLabV3	56	22
ResNet34 + DeepLabV3	**85**	**27**
MobileNetV2 + DeepLabV3	65	54

Table 5. Inference time for 1,024×1,024 input on PyTorch and NVIDIA TensorRT

Implementation of this model on NVIDIA GeForce RTX 2080 using NVIDIA TensorRT requires about thrice less time to process in comparison with PyTorch version of this model on NVIDIA GeForce RTX 2080 Ti. Pre-processing and post-processing operations are also performed on GPU.

5. CONCLUSIONS

Currently, approaches based on convolution neural networks, have achieved significant success in various computer vision tasks, such as image classification, object detection, and semantic segmentation. Architectures of convolution neural networks continue to evolve towards increasing the complexity and performance in terms of accuracy, but it makes them inapplicable to real-time semantic segmentation. In this paper we propose an approach that allows us to use lightweight architectures as a backbone and additional components for real-time solution of semantic segmentation problem for off-road autonomous robotic vehicle. Our approach provides boost of inference time and achieves improvement of segmentation accuracy, which makes it possible to run the semantic segmentation modules in real-time.

ACKNOWLEDGEMENTS

The reported study was funded by RFBR project № 19-07-01248 A.

REFERENCES

V. Badrinarayanan, A. Kendall, R. Cipolla, 2015.
SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017.

A. Buslaev, A. Parinov, E. Khvedchenya, V. I. Iglovikov, A. A. Kalinin, 2018. Albumentations: fast and flexible image augmentations. arXiv:1809.06839v1 [cs.CV].

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, 2016. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 10.1109/TPAMI.2017.2699184.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding // In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv:1706.05587v3 [cs.CV].

M. Gamal, M. Siam, M. Abdel-Razek, 2018. ShuffleSeg: Real-time Semantic Segmentation Network. arXiv:1803.03816v2 [cs.CV].

A. Guha Roy, N. Navab, C. Wachinger, 2018. Concurrent Spatial and Channel Squeeze & Excitation in Fully Convolutional Networks. In: Frangi A., Schnabel J., Davatzikos C., Alberola-López C., Fichtinger G. (eds) Medical Image Computing and Computer Assisted Intervention – MICCAI 2018. MICCAI 2018. Lecture Notes in Computer Science, vol 11070. Springer, Cham.

K. He, X. Zhang, S. Ren, J. Sun, 2015. Deep Residual Learning for Image Recognition // IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

J. Hu, L. Shen, G. Sun, 2017. Squeeze-and-Excitation Networks. // 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 7132-7141.

V. Iglovikov, A. Shvets, 2018. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. arXiv:1801.05746v1 [cs.CV].

J. Long, E. Shelhamer, T. Darrell, 2014. Fully Convolutional Networks for Semantic Segmentation. 10.1109/CVPR.2015.7298965.

N. Ma, X. Zhang, H.-T. Zheng, J. Sun, 2018. ShuffleNet: ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In: Ferrari V., Hebert M., Sminchisescu C., Weiss Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol 11218. Springer, Cham.

O. Ronneberger, P. Fischer, and T. Brox, 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham.

M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, 2018. MobilenetV2: Inverted residuals and linear bottlenecks // 2018 IEEE/CVF Conference on Computer Vision

and Pattern Recognition, Salt Lake City, UT, 2018, pp. 4510-4520.

M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, 2018. RTSeg: Real-time Semantic Segmentation Comparative Study. // 2018 25th IEEE International Conference on Image Processing (ICIP), 1603-1607.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, 2015. Rethinking the inception architecture for computer vision. // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 2818-2826.

M. Tan, Q. V. Le, 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv:1905.11946v3 [cs.LG].

P. Yakubovskiy, 2020. Segmentation Models PyTorch. https://github.com/qubvel/segmentation_models.pytorch