DeepLabv3+ for Semantic Segmentation of Unstructured Rural Environment

Diego Soler[1,2][0000−0002−0357−5251], Mateus Espadoto[1,3][0000−0002−1922−4309], and Roberto Hirata Jr[1,4][0000−0003−3861−7260]

[1] Instituto de Matemática e Estatística - USP, São Paulo - SP, 05508-090, Brazil
[2] dpsoler@usp.br
[3] mespadot@ime.usp.br
[4] hirata@ime.usp.br

Abstract. TAS500v1.1 is a collection of unstructured rural environment images and their respective fine-grained semantic masks. The dataset contains the animal class, and multiple vegetation and terrain classes which are normally not presented on autonomous driving datasets. To train a neural network to the classifying task on this dataset, we defined an image augmentation plan to increase the number of training input images. We only chose methods that would not influence, or confound, important features of the images. We created a testing procedure to evaluate multiple possible networks, changing not only the networks but also the backbone, input dimension and an evaluation method to choose a promising architecture. After choosing the network architecture, we trained it more extensively and achieved a mean Intersection over Union (IoU) of 66.328% on the test dataset. Our custom implementation can be found on: https://github.com/DiegoSoler/custom_keras_segmentation.

Keywords: Semantic Segmentation · Neural Network · Outdoor.

1 Introduction

Autonomous driving is a very prominent field of research. In this environment, a robot has to have a precise knowledge of the environment ahead, which means that its computer vision module has to detect, segment, and classify precisely all classes available in the environment. So autonomous driving networks need extensive training in order to be able to complete its task. There are several datasets for autonomous driving focused on urban environments, such as [6], but there are few datasets for non-urban environments. Metzger et. al., tackled this problem [12], by collecting images using the autonomous vehicle MuCAR-3 [10] driving through unstructured terrain or forest environments.

The TAS500v1.1 dataset is a collection of images and their respective fine-grained semantic masks from images taken through the front windshield view of a car driving in a rural environment. The dataset focuses on vegetation and terrain classes, presenting 23 classes on over 500 scenes.

This technical report describes our approach to train a semantic segmentation model to learn and generate new labels on the TAS500v1.1 dataset.

2 Related Work

Semantic segmentation is a challenging research topic, and multiple solutions have emerged over the years to tackle this problem. Among the possibilities, we can cite SegNet [3], U-Net [13], PSPNet [17] and DeepLabv3+ [5].

Segnet [3] is an encoder/decoder architecture that uses VGG16 as the encoder. The main difference with U-net [13] is that only the pooling indices are transferred along with the network.

U-NET [13] is another neural network architecture for semantic segmentation. The main idea behind the architecture is to find a reduced features' representation of an image, the encoder part, and later expand the representation, the decoder part, resulting in a semantic mask of the parts of the image. The encoder part of the architecture follows a similar part of many convolutional networks, while the decoder consists of upsampling and convolution layers. The entire feature map is passed from the encoder to the decoder part with the same dimension on the U-net architecture.

Differently from the previously mentioned fully convolutional networks, PSPNet, [17], takes into account a global context, which improves its accuracy. The PSPNet encoder is similar to the other networks, but the two last layers are replaced with dilated convolutional layers [15] which helps the encoder to capture more information. Following the encoder, there is a pyramid pooling module that helps learn global information from the image. This module pools from the encoder on different sizes, passing through a convolution and pooling layer, then each size is concatenated after being upsampled to the same size. Finally, the decoder generates the resulting semantic mask.

DeepLabV3 [4] is a semantic segmentation architecture designed by a Google Research group. The architecture employs both a pyramid pooling module and an encoder-decoder architecture, but it improves previous works by adding to the architecture *atrous separable convolutions*. DeepLabV3 outperformed PSPNet in the 2016 ILSVRC Scene Parsing Challenge [14]. DeepLabV3+ [5] improves on DeepLabv3 with a new decoder module that can better predict objects boundaries.

3 Competition

Research on semantic scene segmentation has risen the last years and with the knowledge of what parts of an image refer to what class, autonomous driving vehicles can better understand the environment and make better decisions.

However, most training datasets are focused on structured urban scenarios. To mitigate the restricted dataset problem, Metzger [12] introduced a new dataset focused on unstructured outdoor scenarios. This dataset introduces challenging aspects of the scenes, such as varying lighting and weather conditions. Furthermore, to compare the capabilities of semantics models, the Outdoor Semantic Segmentation Challenge [1] was introduced.

3.1 About

The Outdoor Semantic Segmentation Challenge ran from April 27th, 2021, to August 17th, 2021. The competition's goal was to use a semantic segmentation model on the TAS500v1.1 dataset to predict the semantic mask of one of the 23 classes presented in the dataset. In addition, the models were ranked using the Intersection Over Union (IoU) evaluation metric over the hidden test dataset.

3.2 Dataset Analysis

The TAS500v1.1 dataset consists of 540 images of unstructured outdoor environments alongside their respective fine-grained semantic masks. Each image was cropped and adjusted to a fixed size of 620×2026 pixels. Furthermore, the organizers divided the dataset into 440 images for training and 100 for validation. Only for submission, there were also 100 additional test images (no access to the semantic masks) to rank the competitors' models.

Besides the 23 semantic classes labeled with values between 0 and 22, the dataset has an undefined class (labeled with the value 255) used to mark the image's overexposed regions and named undefined class.

Figure 1, from the competition website, presents a four-column table with, respectively, the class ID for each one of the 24 labels, the class name, the RGB color values, and a descriptive explanation for each class.

3.3 Submission

After the training/validation phase, the competitors were required to use the model to predict the segmentation mask of each one of the 100 images on the test set, save the results in a *.mat* file and submit them to the competition page for evaluation.

The primary metric used to evaluate and rank the models was the Intersection over Union (IoU), also known as the Jaccard index, which measures the similarity of two regions based on their overlap. The IoU is calculated for each class on a test image, based on the hidden semantic mask, then the mean Intersection over Union (mIoU) is calculated as the mean of each IoU score. Another method used for evaluation is the Boundary Jaccard (BJ) [8], which evaluates both the correctly labeled pixel and the similarity of class objects boundaries.

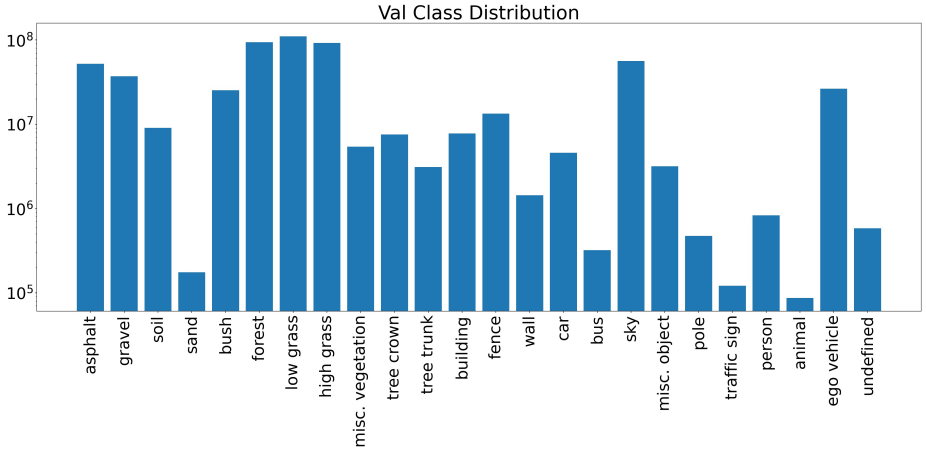
4 Method

In this section, we describe the preprocessing step, the augmentation strategy, and the networks tested.

Class ID	Name	RGB Color	Description
0	asphalt	192,192,192	Drivable surfaces that are made of asphalt: e.g. asphalt roads. This also includes concrete surfaces.
1	gravel	105,105,105	Drivable surfaces that are made of gravel (small stones): e.g. gravel roads.
2	soil	160,82,45	Surfaces that are made of soil: e.g. dirt roads or dirt patches.
3	sand	244,164,96	Surfaces that are made of sand. Sand is sometimes found on the roadside.
4	bush	60,179,113	A bush has several main stems growing from the ground level rather than from one trunk.
5	forest	34,139,34	Multiple trees or bushes, that are part of a forest, thicket or strip of woods.
6	low grass	154,205,50	Grass with a height smaller than about 20cm.
7	high grass	0,128,0	Grass with a height taller than 20 cm.
8	misc. vegetation	0,100,0	Vegetation in the far distance (>300 m) that makes up the scenery.
9	tree crown	0,250,154	The total of an individual plant's above ground parts excluding the tree trunk.
10	tree trunk	139,69,19	The stem and main wooden axis of a tree.
11	building	1,51,73	Larger man-made structures (house, bus stop, skyscraper, transmission towers and construction cranes).
12	fence	190,153,153	A fence is labeled as a surface including any transparent holes that are smaller than 750px.
13	wall	0,132,111	Individual standing wall. Not part of a building.
14	car	0,0,142	Car, jeep, SUV, van with continuous body shape, no other trailers.
15	bus	0,60,100	Bus for 15+ persons, public transport or long distance transport.
16	sky	135,206,250	Open sky, without leaves of trees or any other occlusions.
17	misc. object	128,0,128	Man-made objects and structures that are typically non-drivable.
18	pole	153,153,153	Small mainly vertically oriented pole: e.g. sign pole or traffic light poles.
19	traffic sign	255,255,0	Sign installed for information of the driver in an everyday traffic scene.
20	person	220,20,60	A human or a large crowd of humans.
21	animal	255,182,193	All kinds of animals: e.g. cats, dogs, birds, cows, etc.
22	ego vehicle	220,220,220	Visible parts of the ego vehicle (hood, windshield wiper, etc.).
255	undefined	0,0,0	All unlabeled pixels are defined with the label undefined. In most cases this includes overexposed areas of the image.

Fig. 1: Table from the competition website [1] describing each semantic class. The columns present the class ID for each one of the 24 labels, the class name, the RGB color values, and a descriptive explanation for each class.

(a) Histogram of the class distribution on the 440 training dataset images. The y-axis is in logarithmic scale.



(b) Histogram of the class distribution on the 100 validation dataset images. The y-axis is in logarithmic scale.

4.1 Preprocessing

The undefined class consists of camera overexposure and water on the image. It consists of the value 255 on the semantic mask, so our first step was to decide the best approach to deal with the undefined class. After some literature review, we decided to test two methods: ignore this class by using the weighted cross-entropy loss and setting the undefined class weight to zero; change the class value to 23 and train the network with 24 classes rather than 23.

After some preliminary tests with the semantic models, we chose to use the second approach. The rationale is because some data augmentation strategies change the images geometrically, needing some pixels on the semantic mask to be filled in. What was found is that predicting these regions resulted in better results than ignoring them. Therefore, as the first preprocessing step, all semantic mask values of 255 were changed to 23. After predicting the test images, the labels needed to be changed back to 255.

4.2 Augmentation

Data augmentation is a successful approach to improving a model. The idea is to increase the number of images in the training dataset by simple image transforms and significantly improve the model's accuracy. We used the library imgaug [11], which is equipped with a collection of image augmentation methods that can be composed to create new methods.

Some of the transformations, specifically the geometrical transformation, required parts of the image to be completed. The default values for both semantic mask and source are 0, the label for the "asphalt". Therefore, we set this value to 23, which is one of the reasons we found it better not to ignore the undefined class but rather use it.

The image augmentation methods used are summarized in Table 1 and detailed in the following subsections.

Table 1: Augmentation transformation used for training. All augmentation transformation used are from the imgaug library.

Augmentation	**Input Parameter**	**Fill Method**
Rotation	-25 to 25 degrees	fill with 23
Scale (XY)	0.8 to 1.2	fill with 23
Translation (XY)	-20px to 20px	fill with 23
Flip	vertical axis	-
Gamma Constrast	0.5 to 2.0	-
Median Blur	1px to 3px	-

Rotation This transformation rotates the image around the center of the image, filling the image with a pixel value of 23. We used the parameters for rotations between -25 to 25 degrees. This means that each image will be rotated by a random angle in this interval.

Scale This transformation scales the image from the center of the image. We used both X and Y scales and the parameters used for scales are between 0.8 and 1.2.

Translation This transformation shifts the image horizontally, or vertically, a certain amount of pixels, filling the pixels with a pixel value 23. We used both X and Y translation and the parameters used for translation are between -20 pixels and 20 pixels.

Flip This transformation flips the image. We only used vertical flip and a parameter of 0.5, meaning there is a probability that the image will be flipped of 50%.

Gamma Contrast This transformation modifies the contrast of an image according to the formula $255 * ((v/255) ** gamma)$, where v is the pixel value. The parameters used are between 0.5 to 2.0.

Median Blur This transformation blurs the image with a median window. The parameters used are between 1 to 3 pixels, then the image will be blurred with a window size of one of 1x1, 2x2 and 3x3.

4.3 Semantic Segmentation

Our solution uses the Image Segmentation Keras library [9] and takes advantage that most of the architectures we planned to test are ready on the library. The list of available architectures and some other library features are presented on the Image Segmentation Keras Github page.

We also used the library's mIoU evaluation and the implemented method to generate images with overlaid prediction. Both functions help in evaluating the network performance after training.

4.4 Network targeting

We selected three semantic segmentation architectures from the Keras' library and tested them on the TAS500v1.1 dataset: SegNet, U-Net, and PSPNet. Besides those architectures, we also added DeepLabV3+ [16] to our Keras' library installation.

Another essential strategy to improve accuracy is to use a pre-trained network as the encoder of the architecture, also known as the backbone of the model. The architectures used and tested were: ResNet, MobileNet, VGG-16, and Xception. Those networks were extensively trained on massive datasets such as ImageNet [14], PASCAL-VOC [7] and Cityscapes [6]

Two standard procedures are possible: (1) transfer learning, which is copying the trained weights and locking the networks for no further training, or (2) finetuning the networks, which does not lock the networks, allowing for new weight during training. The former method is better when it is known that the goal dataset is similar to the pre-trained one. As the TAS500v1.1 dataset is not similar to those used during training the backbones, the group decided on finetuning the network.

Table 2 presents a comparison between the tested architectures. The mIoU score was computed on the validation split of the dataset. The training procedure was the same for all networks, with only the input dimension changing, the standard input dimension of each model's backbone.

Table 2: Table comparing multiple segmentation models and backbone combinations. The models were trained with the standard input dimension for 10 epochs.

Segmentation Model	**Backbone**	**Trained Dataset**	**Input Dimension**	**Val mIoU**
Segnet	VGG 16	imagenet	(416, 608)	0,376
Segnet	Resnet-50	imagenet	(416, 608)	0,358
Segnet	MobileNet	imagenet	(224, 224)	0,469
U-Net	VGG 16	imagenet	(416, 608)	0,422
U-Net	Resnet-50	imagenet	(416, 608)	0,466
U-Net	MobileNet	imagenet	(224, 224)	0,513
PSPNet	VGG 16	imagenet	(384, 576)	0,365
PSPNet	Resnet-50	imagenet	(384, 576)	0,486
DeepLabV3+	**MobileNet**	**pascal_voc**	**(512, 512)**	**0,529**

DeepLabV3+ architecture achieved the best mIoU score, followed by U-Net with the MobileNet backbone, which is interesting, as PSPNet has a higher rank in another competition. One reason is that being a larger network, PSPNet needs more training to achieve better results, while U-Net is easier to train. Nevertheless, the group decided on using the DeepLabV3+ model.

Figure 2 shows the output of the U-net overlayed on the original image, while Fig. 3 shows the output of DeepLabv3+, which is less pixelated and has fewer false positives.

5 Results

DeepLabV3+ was the network chosen to generate the semantic mask for our participation in the competition. This architecture achieved a high rank on the

Fig. 2: U-net prediction overlayed on the original image

Fig. 3: DeepLabV3+ prediction overlayed on the original image

Pascal VOC semantic segmentation competition [7] and, as presented in Table 2, had the best initial performance on the competition dataset. This section presents the fine-tuning of the chosen architecture input parameters to reach the best mIoU score possible.

5.1 Input Resolution

Input resolution impacts the performance of a model [2] and, during testing, one of the possible adjustments we tested was changing the input resolution. Table 3 presents the impact of changing the network's input resolution. Besides resolution, Table 3 presents a combination of two other variations: the backbone model and the dataset used for training. All models have been trained for ten epochs, and the last column presents the mIoU on the validation dataset.

The results show that downsampling can impact the performance of the model in several aspects, as expected. First, lowering the resolution lowers the number of available features and can help the performance. However, the downsampling can not be so harsh. On the other hand, the higher the input dimension, the better the improvement in the results is at the cost of increasing training time and the number of epochs. After extensive testing, the dimension that gave the best results and whose training was still viable was at half of the original resolution, i.e., 310×1013 pixels. Other resolutions used had a aspect ratio (width/height) between 1 and 1.5, where the original image has aspect ratio has

Table 3: Table comparing the chosen architecture, DeepLabV3+ with different input resolution, backbone, and pre-trained dataset. The mIoU reported refers to the validation split of the TAS500 dataset. The models were trained for 10 epochs.

Backbone	**Dataset for Pretraining**	**Input Dimension**	**mIoU**
MobileNet	pascal_voc	(512, 512)	0.529
MobileNet	pascal_voc	(224, 224)	0.476
MobileNet	pascal_voc	(310, 1013)	0.590
MobileNet	cityscapes	(310, 1013)	0.602
Xception	pascal_voc	(310, 1013)	0.595
Xception	cityscapes	(310, 1013)	0.611

3.27, which means that the images were greatly morphed before the network, which also can explain the improvement in using the chosen resolution.

5.2 Backbone

During the test presented in Table 3, the group also wanted to know what backbone improved the network the best, so the training was repeated for two backbone architecture, MobileNet, which helped U-Net to achieve the best mIoU on Table 2, and Xception, which is the backbone used on Pascal Voc segmentation competition. The Xception backbone improved the mIoU to 0.611.

Another critical parameter is the dataset used to train the backbone. Among the options, Cityscapes and PASCAL VOC were explored. The general rule of thumb is to use a model trained on a dataset similar to the goal dataset. In this case, the dataset is an unstructured outdoor dataset, different from PASCAL VOC and Cityscapes, which are structured, so there is no clear best candidate, which is also a good reason to finetune the network. The best results were achieved using Cityscapes, as seen in Table 3.

5.3 Network training

All networks tested used the same training configuration. The loss used was categorical cross-entropy and the optimizer used was the Adam optimizer with standard configuration, (learning_rate=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-07), the training parameters used were:

- Epochs: 10
- Steps per epoch: 1024
- Batch Size: 4

With this configuration, the group reached a mIoU on the test dataset of 61.658%. After this result, we adjusted the training parameters with the goal to train the network longer with more augmented images in order to reach our second place model with a mIoU on the competition test set of 66.328%. the training parameters used are:

- Epochs: 25
- Steps per epoch: 2048
- Batch Size: 6

5.4 Discussion

Our final network had a validation mIoU of 64.443% and a test mIoU of 66.328%. Table 4 presents the IoU of each class. The classes sand, traffic sign, and undefined are clearly difficult for the network. Those classes are also the less represented classes, losing only to the animal class. The solution to this problem could be using a weighted categorical cross-entropy loss function to improve the weights of those classes. Figure 4 presents two good examples, and Fig. 5 two bad examples where the network fails, on the top row we have a sand pile on the bounding box that the network labeled as a mixture of other classes, mainly ground classes, and on the bottom row the bounding box shows a traffic sign that is mislabeled as asphalt.

Table 4: Table presenting the IoU on the validation dataset for each of the 23+1 classes.

asphalt	**gravel**	**soil**	**sand**	**bush**	**forest**
0.9146	0.8472	0.5937	0.0000	0.6082	0.8260
low grass	**high grass**	**misc. vegetation**	**tree crown**	**tree trunk**	**building**
0.8364	0.8139	0.9372	0.7270	0.4488	0.7427
fence	**wall**	**car**	**bus**	**sky**	**misc. object**
0.7054	0.1516	0.8845	0.6744	0.9663	0.4042
pole	**traffic sign**	**person**	**animal**	**ego vehicle**	**undefined**
0.4514	0.1247	0.5674	0.6402	0.9559	0.0558

6 Conclusion

In this work, we explained our approach to using semantic segmentation methods to generate semantic masks for the TAS500v1.1, which contains images of unstructured outdoor environments. We tested multiple semantic segmentation deep learning architectures to face the problem. We tested multiple semantic segmentation deep learning architectures to face the problem. Our final solution uses the DeepLabV3+, end-to-end trainable, and we achieved a mIoU of 66.328% on the test dataset.

The competition was a good opportunity to learn more and understand better the networks models for semantic segmentation applied to an unstructured and unbalanced non-urban dataset. In the future, we think that another possibility is using Vision Transformers, which has already improved classification networks.

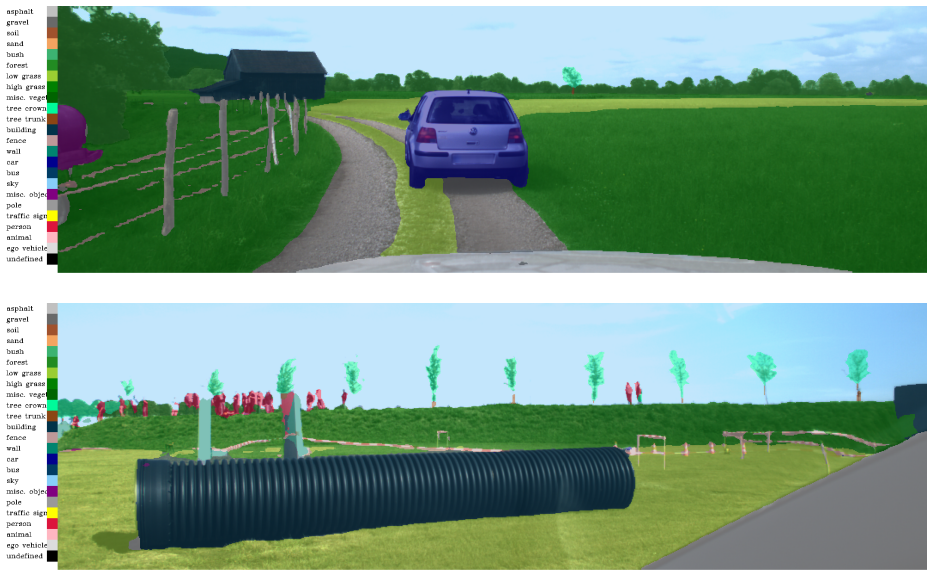
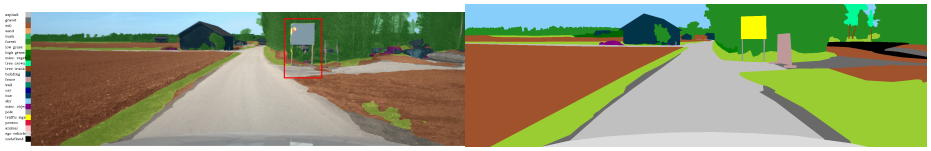


Fig. 4: Good Examples: The network manages to predict the fine details of both images.

(a) Prediction (b) Groud Truth



(c) Prediction (d) Groud Truth

Fig. 5: Bad Examples: Network failing to predict two under represented classes. Sand, first row, and Traffic Sign, bottom row.

References

1. Outdoor semantic segmentation challenge (dagm gcpr 2021). https://competitions.codalab.org/competitions/31086, accessed: 2021-9-15
2. Abello, A, A., Jr, H., R., Wang, Zhangyang: Dissecting the high-frequency bias in convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021. pp. 863–871. IEEE (2021)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional Encoder-Decoder architecture for image segmentation (Nov 2015)
4. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation (Jun 2017)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-Decoder with atrous separable convolution for semantic image segmentation (Feb 2018)
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
7. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision **111**(1), 98–136 (Jan 2015)
8. Fernandez-Moral, E., Martins, R., Wolf, D., Rives, P.: A new metric for evaluating semantic segmentation: Leveraging global and contour accuracy. In: 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE (Jun 2018)
9. Gupta, D.: image-segmentation-keras: Implementation of segnet, FCN, UNet , PSPNet and other models in keras
10. Himmelsbach, M., Luettel, T., Hecker, F., Hundelshausen, F., Wuensche, H.J.: Autonomous off-road navigation for mucar-3. KI - Künstliche Intelligenz **25**(2), 145–149 (2011). https://doi.org/10.1007/s13218-011-0091-1
11. Jung, A.B., Wada, K., Crall, J., Tanaka, S., Graving, J., Reinders, C., Yadav, S., Banerjee, J., Vecsei, G., Kraft, A., Rui, Z., Borovec, J., Vallentin, C., Zhydenko, S., Pfeiffer, K., Cook, B., Fernández, I., De Rainville, F.M., Weng, C.H., Ayala-Acevedo, A., Meudec, R., Laporte, M., et al.: imgaug. https://github.com/aleju/imgaug (2020), online; accessed 01-Feb-2020
12. Metzger, K.A., Mortimer, P., Wuensche, H.J.: A Fine-Grained dataset and its efficient semantic segmentation for unstructured driving scenarios. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 7892–7899 (Jan 2021)
13. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation (May 2015)
14. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
15. Yu, F., Koltun, V., Funkhouser, T.: Dilated residual networks (May 2017)
16. Zakirov, E.: Keras implementation of deeplab v3+ with pretrained weights. https://github.com/bonlime/keras-deeplab-v3-plus
17. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network (Dec 2016)