Learning to Identify the Navigable Path in Unstructured and Off-Road Scenes from a Single Dataset

André Victória Matias[1 [0000−0003−0268−0233]], Juliana Marian Arrais[2 [0000−0002−3903−2646]], Allan Cerentini[1 [0000−0001−8979−4983]], Bruno Juncklaus Martins[1 [0000−0001−6155−0557]], Felipe Trindade Radovanovic[3 [0000−0002−2925−4368]], Gabriel Machado[1 [0000−0003−2779−7585]], João Gustavo Atkinson Amorim[1 [0000−0003−3361−6891]], and Aldo von Wangenheim[4 [0000−0003−4532−1417]]

[1] Department of Informatics and Statistics, Federal University of Santa Catarina, Florianópolis, Brazil
{andre.v.matias, allan.c, bruno.juncklaus, gabriel.g.m.machado, joao.atkinson}@posgrad.ufsc.br
[2] Department of Physics, Federal University of Santa Catarina, Florianópolis, Brazil
juliana.arrais@grad.ufsc.br
[3] Department of Electrical and Electronic Engineering, Federal University of Santa Catarina, Florianópolis, Brazil
felipe.radovanovic@grad.ufsc.br
[4] Brazilian Institute for Digital Convergence, Federal University of Santa Catarina, Florianópolis, Brazil
aldo.vw@ufsc.br

Abstract. One of the most important tasks in a visual perception system for automotive navigation and Navigable Path Detection (NPD) is the perception of the environment and detection of obstacles. This task is a challenge still to be overcome for navigation on heavily damaged and unpaved roads. To address this, we applied methods previously developed by us, in particular multi-resolution methods and successive training methods, and new, state-of-the-art, multi-resolution methods using only the TAS500 dataset in the context of the DAGM GCPR 2021 Outdoor Semantic Segmentation Challenge. As our multiresolution approach, we investigated a classical U-Net architecture using the one cycle training policy and an adaptation of a successive growing resolution training strategy. As a second solution, we employed the *HRNet* model, which has an architecture that naturally learns multiple resolutions during the training, in two versions: the original implementation and a refined Semantic Segmentation version, called *SemTorch*, which also employs the one cycle training policy. The results show that *SemTorch HRNets* have the potential to be architectures-of-choice for embedded navigable path detection as they achieved 0.659 mIoU in the test set (third place in the challenge) and 15.74 FPS in a TensorRT engine on an NVIDIA JETSON TX2.

Keywords: Deep Learning · Navigable Path Detection (NPD).

1 Introduction

One of the most important tasks in a visual perception systems for automotive navigation and Navigable Path Detection (NPD) is the perception of the environment and the obstacles contained in this environment. This is important not only for the vehicle to navigate adequately, but also to avoid accidents. Besides detecting if there is a navigable path, where it is and which kind of roughness it presents, a robust intelligent NPD system must be able to collect the necessary information for the detection and tracking of obstacles, such as: position, size and relative speed that each detected obstacle presents [4,14,13,15]. The task of reliable path and obstacle detection is a challenge still to be overcome in the field of visual perception for vehicle and robotic navigation on heavily damaged and unpaved roads. The vast majority of the research works in the literature is based upon roads in good condition from developed countries. They do not consider varying road surface types and damages along the road surface and even fewer situations presenting surface damages [17].

Besides, many vision-based obstacle detection methods aim to identify only certain types of immediate threats, such as cars, animals, bicycles or pedestrians. This can lead to systems that either present a considerable number of false positive detection alarms or that miss some kinds of conditionally navigable obstacles, such as bumpers, cracks, potholes and puddles, that should be detected and dealt with [27]. Paths lacking horizontal and vertical signage, such as dirt roads, forest paths and off-road tracks, present special challenges for Passive Vision-based Navigation (PVN) systems because all information about the navigability of the path ahead has to be inferred from visual input [15].

1.1 Contextualization

We understand that situations commonly found in in-development countries, such as Brazil or India, like dirt roads and suburban areas with extremely damaged paved roads, presenting potholes, puddles, speed bumpers and cracks, have to be addressed by research on autonomous navigation systems and ADAS [15]. Such countries present large markets for autonomous navigation systems, not only for personal vehicles, but also for goods transportation and intelligent agricultural implements. In countries like Brazil, it is not uncommon that an expedition company has routes that encompass hundreds or even thousands of kilometers of dirt roads or extremely badly maintained paved roads. In this context, the automated identification of elements in unstructured and unpaved road scenes is of extreme importance.

Our research group has been performing research on segmentation-based techniques for the identification of the navigable path for autonomous vehicles on dirt roads for some time now, both employing classic computer vision [26,15] and Convolutional Neural Networks (CNN) [17,16] . We also have been performing applied computer vision research in other application domains such as oil exploration [2] and medical image processing [12] . In order to tackle the present challenge, we employed the experience and the evidence gathered in these works, together with the requirements we defined below.

1.2 Objectives

Motivated by the results obtained in our previous works and by the lack of resources to gather additional data or to adapt and segment manually external datasets in order to make them compatible to the classes used in this challenge, we decided to investigate the following research question: What is the best segmentation result that can be obtained employing only the given dataset as a training input?

For this purpose, we stated the following objectives:

(i) Test on this dataset methods previously developed by us, in particular multi-resolution methods and successive training methods;
(ii) Test new, state-of-the-art, multi-resolution methods on this data set

Following these objectives, we ended up with three different experiment setups, which are described below.

2 Material and Methods

2.1 Class Weights

Following the evidence gathered in [17], we decided to apply weights to the individual classes accordingly to their distribution in the dataset. In order to define weights for the different classes of the dataset, we based ourselves upon the class distribution graph[5] provided on the challenge site. The weight of each class w_c, excluding the "undefined" is calculated based on the number of pixels of the asphalt class p_a and number of pixels of the class p_c as follows: $w_c \approx log_{10}p_a/log_{10}p_c$. This resulted in the weights shown in table 1. We selected the asphalt class as the reference class since the main objective of an autonomous vehicle is to detect the navigable path and this class is one of the most common class in the dataset, providing a natural divider.

We employed these weights in different manners in all three experiments described below.

2.2 UNets

For our first approach we selected as our Semantic Segmentation (SS) approach the U-Net architecture [18] for segmentation as it has shown state-of-the-art performance at this task on various image modalities, such as medical image segmentation [1,11,25,23,12], segmentation of rock samples in the paleontological field [2] and also navigable path detection in unstructured domains [17]. A UNet model is composed by two main components:

[5] http://https://www.mucar3.de/icpr2020-tas500/images/barchart.png

Table 1. Weights calculated from the classes distribution

Weight and Class name			
Weight	**Class Name**	**Weight**	**Class Name**
1.00	asphalt	1.04	gravel
1.21	soil	1.56	sand
0.96	bush	0.97	forest
1.00	low grass	1.00	high grass
1.16	misc. vegetation	1.16	tree crown
1.40	tree trunk	1.14	building
1.10	fence	1.30	wall
1.18	car	1.48	bus
1.00	sky	1.28	misc. object
1.45	pole	1.70	traffic sign
1.36	person	1.70	animal
1.05	ego vehicle	0.00	undefined

- A SS-framework, which is a Deep Learning (DL) structure in "U"-form, composed of two parts: a descending *encoder* arm, that takes pixel information and learns ("codifies") an abstract representation of pixel distribution in each class, and an ascending *decoder* arm that generates an image where pixels are labeled accordingly to their classes.
- A specific Image Classification (IC) DL model as the processing and learning modules of both arms, called the *backbone*.

In this experiment, the IC models used as backbones were the ResNet-34 and ResNet-50 [5] with an ADAM optimizer [10] and the cross entropy loss function. ResNets are state-of-the-art IC models commonly employed for SS tasks. Additionally, the U-Net framework provides extra horizontal connections between corresponding layers of the instances of the backbone in each arm, which learn to control the different levels of abstraction of the SS process. We employed the high-level UNet/ResNet implementation provided by the *fastai* API [7].

Strategy #1 - Multiple Training, Crescent Resolution We applied the *One Cycle Policy* as described in [20] to optimize learning rates during training for segmentation. Additionally, we trained each segmentation model iteratively with 155x506 (1/4 size), 310x1013 (1/2 size) and 620x2026 (full size) versions of the train set images, in this order. We performed training in two stages for each resolution, with a transfer-learning and a fine-tuning-learning cycle each time: the first with the weights of the internal encoder layers frozen (transfer learning stage) and the second with all internal encoder layers unfrozen (fine tuning stage). After each training conclusion, the weights of the previous image scale were used to initialize the model for the next training. To find the best maximum learning rate for each training step for segmentation, we employed the method discussed in [19]. This iterative training method with progressive resolution is an adaptation of the strategy presented in [6] and has been successfully used in difficult-to-train datasets such as micro-tomographic volumes in the Marine Paleontology field [2] and in the Cytology field [12]. Figure 1 shows a diagram of this training process.

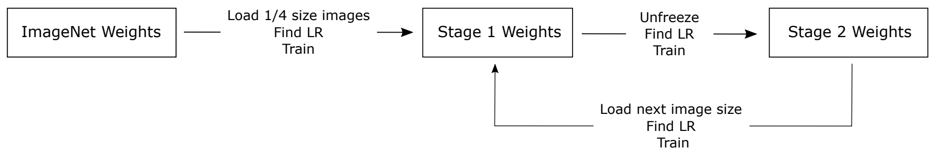

Fig. 1. Multiple Training, Crescent Resolution strategy diagram.

2.3 HRNets

The HRNet (High Resolution Neural Network) [8] was chosen as our second approach. This new CNN model presents a simultaneous multi-resolution learning approach and also enables for easy changes to their architecture, allowing to adapt it to a specific task, such as IC, OD or SS [21,24,28] (see figure 2). Additionally, the HRNet [8] presents a mathematically elegant and homogeneous structure, which we understand is promising for further improvements.

Image Classification (IC): The neural network classifies the image based on the image's primary object classes [8,21]. These networks are usually more accurate in categorizing and detecting items in photos, not to mention the position and quantity of these objects, and this is the best solution when you don't need to cut an object or know its position in the image.

Object Detection (OD): The neural network recognizes the objects in the image and indicates their location by superimposing a convex hull, usually a rectangular frame, around their estimated location [24]. These networks do not have the same classification performance as IC networks since they make more mistakes, but they indicate where the objects are in the image and how many instances of each category are present.

Semantic Segmentation (SS): the SS HRNet architecture was the approach we used in this experiment (see figure 2). The CNN classifies individual pixels in an image according to the object class that each pixel belongs, thereby segmenting the image into areas, each of which represents an object. These object detection networks have lower classification performance than OD networks, but, in medical applications and autonomous driving, for example, these networks are often essential because they allow for the exact segmentation, localization and measurement of objects-of-interest in the image.

HRNets attracted little attention from the international scientific community after its release in 2019, but are now being evaluated as a state-of-the-art model. We employed HRNets in this work in the context of two different strategies:

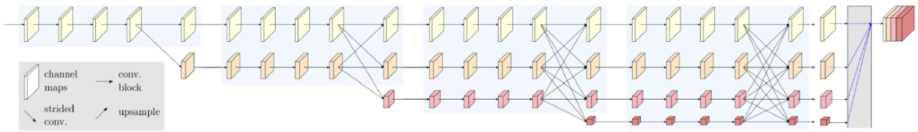

Fig. 2. HRNet for Semantic Segmentation (SS): "network head", shown in the gray rectangle to the right of the figure, has an upsampling neural structure in one information step, which ends with a set of segmentation presentation layers consisting of a convolutional block that outputs an image of labeled pixels [28].

Strategy #2 - Single Training, Original HRNet Approach In this first experiment with the HRNet model, we employed the original implementation of the HRNet model, in its refined HRNet OCR for Semantic Segmentation version, as published by their authors in [9]. This HRNet implementation was in PyTorch and also used the ADAM optimizer, but does not employ the One Cycle training policy. We tested the HRNet 18, 32 and 48 models. All models were run on NVDIDIA P100 GPUs. Since this model is supposed to learn a multi-resolution representation of the images, we trained the network using a standard single cycle of the transfer-learning/fine-tuning workflow. The network was fed with images in the original 620x2026 (full size) resolution.

Strategy #3 - Single Training Training, HRNet with One Cycle Policy In this second experiment with the HRNet model, we employed the refined implementation of the HRNet model, as published in [3]. This HRNet implementation was developed using the same *fastai* [7] high-level library as the UNet model described previously and employed both the ADAM optimizer and the One Cycle training policy. Here, we also tested the HRNet 18, 32 and 48 models. All models were run on NVDIDIA P100 GPUs. This networks we also trained using a standard transfer-learning/fine-tuning workflow. The network was also fed with images in the original 620x2026 resolution.

3 Results

The best results of each combination of models and training strategies are presented in Table 2. The table presents also the reference implementations[6] of each network we employed in each of the experiments.

For the HRNet32 model, which has shown the best results on the validation set, we also performed two additional experiments including augmented training set versions. For the first experiment we included the 20 top losses samples from the validation set into the training set and in the second experiment we evaluated

[6] The code used to reproduce the results is available at https://codigos.ufsc.br/lapix/ossc.

the test set and segmented manually the ten images that presented top losses and also included these into the training set. This last procedure was against the rules of the challenge and was not officially submitted to the competition, but it presents an interesting result that we will discuss below, in section 4.

Table 2. Best results for each employed strategy (with reference implementation)

Model/Strategy	**Val IoU**	**Test IoU**	**Epochs**
UNet w/ ResNet 34 [7] Multiple Training, Crescent Resolution	0.333	0.342	121
HRNet 18 OCR [9] Original Strategy	0.612	0.598	600
HRNet 32 [3] One Cycle Policy (Train Only)	0.646	0.645	107
HRNet 32 [3] One Cycle Policy (Train + 20 Val)*	0.759	0.659	84
HRNet 32 [3] One Cycle Policy (Train + 20 Val + 15 Test)**	**0.792**	**0.671**	95

*These results were obtained including an augmented subset of the validation set into the training data (see discussion in the text).
**These results were obtained including a manually segmented subset of the test set into the training data. This extra labels are available at
https://arquivos.ufsc.br/d/653ce471bdac483ea222/.

For all models a very fast overfitting behavior during training could be observed, as exemplified in the training curve for the HRNet 18 OCR model, in figure 3. We performed an *ad hoc* visual inspection of the segmentation results and it indicated that this was related to (a) the difficulty in learning some classes, such as *sand*, and (b) the high degree of similarity between vegetation classes, which led to a large number of miss-classifications that could not be correctly learned. We will also discuss this further in section 4.

The original HRNet OCR implementations, which do not employ the One Cycle Policy, have shown a much slower learning performance and also poorer results than the same models extended with this meta-optimization strategy. This is also true when comparing the original HRNet implementation to our UNet/ResNet that also uses the One Cycle Policy.

Figure 4 shows some exemplar results of some of the architectures we tested: HRNet 18 OCR [9], UNet w/ ResNet 34 [7] and HRNet 32 [3] employing the one cycle policy and a custom training dataset composed of the original training set plus the 20 images from the validation set with the largest errors. We chose some good and some bad results based upon visual inspection of the resulting segmentation. All results are available online [7].

[7] http://www.incod.ufsc.br/dagm-ossc/

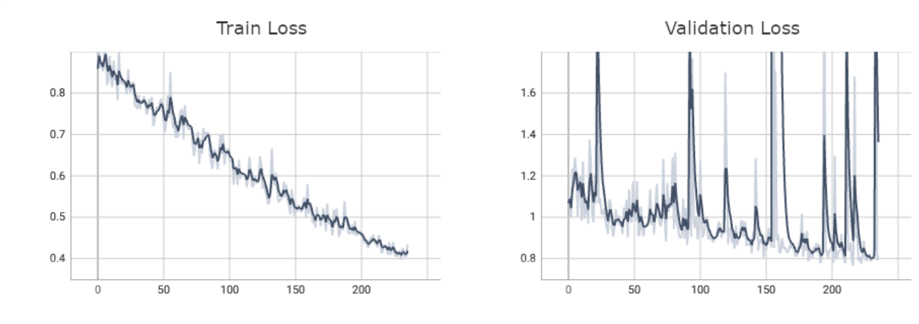

Fig. 3. Train versus validation loss graph for the HRNet 18 OCR model.

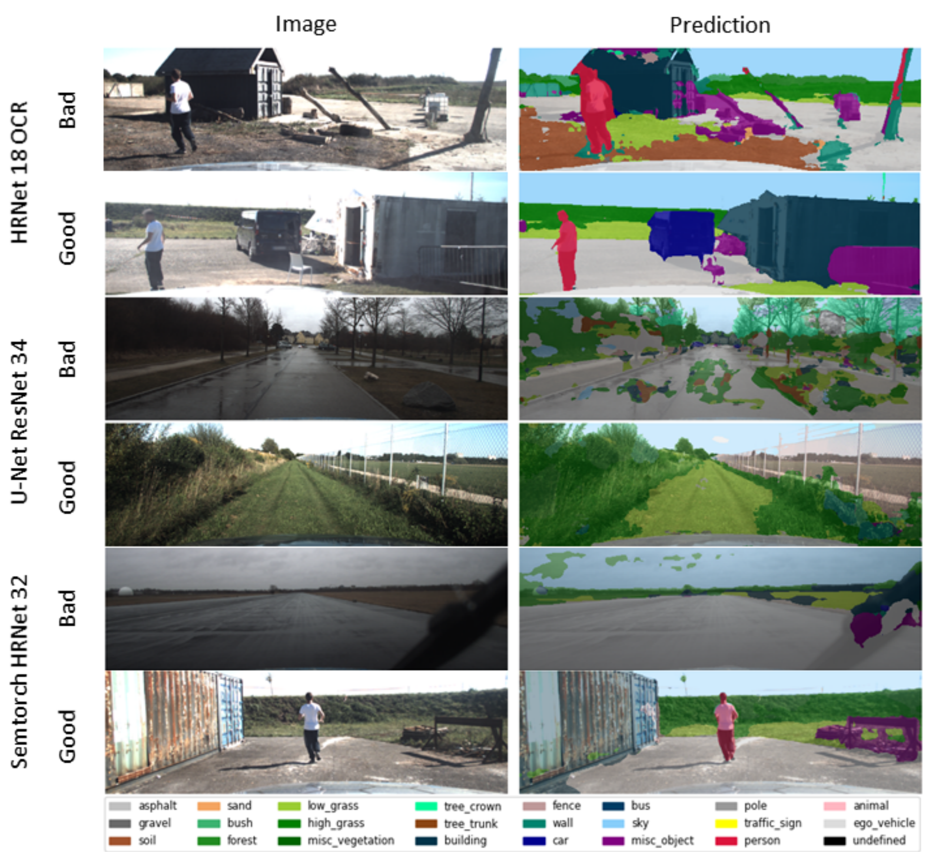

Fig. 4. Good and bad exemplar results of some of the architectures we tested.

4 Discussion and Conclusions

Due to limited resources we decided not to adapt other dirt-road datasets, such as our own RTK Dataset [13], to extend the training data, and to work only with the 400 training and 100 validation images provided in the challenge and to focus on identifying a CNN architecture that could produce effective results with only these limited data. Thus, in our study we explored which results could be obtained for the task of navigable path detection with semantic segmentation employing a multi-resolution approach and using only the dataset provided by the competition as a training input.

As our multiresolution approach we investigated two different solutions: First, a classical U-Net architecture, refined with the one cycle training policy [20], and an adaptation of the successive growing resolution training strategy [6], which we empirically adapted for semantic segmentation and refined into a six step process (see figure 1). This was a strategy that we have employed with great success elsewhere and we were confident that it would work well here [2,17,23,12]. As a second solution, we considered the use of a new neural network model, which possesses an architecture that naturally learns multiple resolutions during the training, the *HRNet* [24]. This model we employed in two versions: the original implementation proposed by the authors and a refined SS version, called *SemTorch* [3], which also additionally employs the one cycle training policy [20]. In this context, the High Resolution Neural Networks [21,24,28,8], especially when extended with training strategies as the One Cycle Policy, have shown promising results.

The observed fast overfitting behavior could be justified by many factors. One is surely the small dataset allied to the complexity of the data. Some classes, such as *sand*, are so underrepresented that, e.g., an HRNet 18 OCR never learned an IoU greater than 0.0 for this class before overfitting, unless extreme weights were set for the class, which lowered the network's performance for other classes. This is a problem that a larger dataset surely will solve, since sand is something that is commonly found on dirt roads. A different problem were the many different vegetation classes: here we could observe not only many misclassifications, but also a tendency of *segment leakage* between vegetation classes, leading to low IoUs for these classes, even if parts of the vegetation areas were correctly classified. Here we question the usefulness of these fine-grained vegetation classes: we understand that, from the point of view of an autonomous car, there are only two semantics for vegetation: *navigable vegetation*, such as roadside grass and fields, and *non-navigable vegetation*, such as bushes, trees, high grass and hedges, which will represent an obstacle. Figure 5 shows some examples of segment leakage on results obtained with the SemTorch implementation of the HRNet 32. The fusion of vegetation classes whose differentiation is not relevant to a vehicle, into only two classes, *navigable* and *non-navigable* vegetation could enhance the learnability of the dataset.

Threats to Validity This work was performed in the context of a competition, therefore we did not have time nor resources to revise the code of the implementations of the two HRNet interpretations we used (original and SemTorch) nor perform extensive experiments with hyperparameters. In this context, we accepted the assumption that the main difference in the training speed and accuracy that could be observed between the two implementations was due to the use of the One Cycle training policy, which is the only explicit technical difference between both interpretations of the model, but both implementations actually differ in more points: the original HRNet implementation uses custom Python versions of some basic mathematical functions and also has custom CUDA implementations of weight adjustment routines, whereas the SemNet interpretation employs many fastai optimizations for standard PyTorch routines.

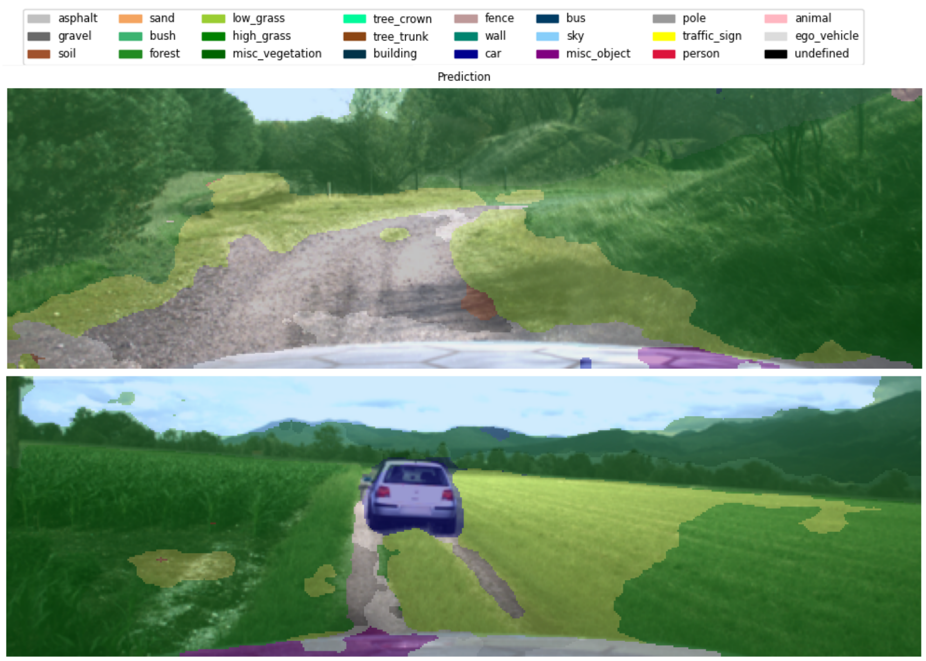

Fig. 5. Examples of segment leakage for different classes of vegetation (HRNet 32).

5 Future Work

The final purpose of the task of navigable path detection using semantic segmentation is the real-time identification of roads and obstacles in autonomous vehicles. This means that a necessary requirement for any neural network performing this task is to be usable in the context of an embedded system. This in turn, raises the question if the models we investigated in this study will perform adequately in a real-time situation and on a hardware platform suitable to be embedded into a vehicle. In order to test this, we performed a small experiment on an NVIDIA JETSON TX2 platform [22], which we describe below, as a preview of research to come.

In this sense, we decided to train new models using the implementation from SemTorch with the RTK Dataset [13]. Afterwards, we evaluated its efficiency by comparing them with previous models of U-Net with ResNet-34 and U-Net with ResNet-18, both trained with the same dataset. For this experiment we employed the RTK dataset and not the challenge material because we wanted a set of consecutive images extracted from a video in order to realistically simulate a real-time segmentation and because the U-Net related results were already available for the RTK dataset from previous experiments.

Initially, we faced a counter intuitive behavior, in which the time of training of the HRNet implementation was substantially faster than U-Net with ResNet. Requiring even less epochs to reach its optimal training point. The HRNet models were trained with only 50 epochs, while the UNet models required 200 epochs, both splitted into two stages (first half with unbalanced weights and the second half with balanced weights).

On table 3 we present the results in terms of weighted accuracy for the RTK Dataset. One can note that with this new architecture, we were able to improve the accuracy in almost 3%, reaching 98.23% with the HRNet 32. And yet, the model's sizes were dramatically reduced, so that with the HRNet-18 occupied only 36.8 MB while preserving 97.98% of accuracy. When we compare the HRNet-18 with ResNet-18, the two smallest models for each architecture tested here, the first one improved the accuracy in 1.86% and reduced disk occupation in 69%.

Subsequently, we embedded all models in an NVIDIA Jetson TX2, in order to reproduce a real-time application with limited graphical resources for these architectures. Once the models were trained, they were converted from PyTorch to ONNX format. And, finally, on the TX2 we optimized they with TensorRT engine, using the version 7.1.3. The performance in frames per second (FPS) of HRNets were about 300% superior than the implementation of U-Net with ResNet, reaching an average of 21.54 FPS with HRNet-18 against 7.75 FPS from U-Net with ResNet-18, as presented in table 3.

Table 3. Weighted accuracy results of models trained and tested in the RTK Dataset with a TensorRT engine on an NVIDIA JETSON TX2

Model	**Performance**	**Accuracy**	**Size**
UNet w/ ResNet 34	7.24 FPS	95.95%	157.2 MB
UNet w/ ResNet 18	7.75 FPS	96.12%	118.7 MB
HRNet 32	15.74 FPS	98.23%	112.7 MB
HRNet 30	16.24 FPS	98.20%	99.3 MB
HRNet 18	21.54 FPS	97.98%	36.8 MB

These results show that HRNets, in the form of the refined SemTorch version, have potential to be architectures-of-choice for embedded navigable path detection and deserve further study and refinement. In this sense, the next steps for future works will be focused on the implementation of HRNet models for real-time applications and environments.

References

1. Atkinson Amorim, J.G., Buschetto Macarini, L.A., Victória Matias, A., Cerentini, A., Botelho De Miranda Onofre, F., Sherlley Casimiro Onofre, A., Von Wangenheim, A.: A novel approach on segmentation of agnor-stained cytology images using deep learning. In: 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS). pp. 552–557 (2020). https://doi.org/10.1109/CBMS49503.2020.00110
2. Carvalho, L., Fauth, G., Baecker Fauth, S., Krahl, G., Moreira, A., Fernandes, C., von Wangenheim, A.: Automated microfossil identification and segmentation using a deep learning approach. Marine Micropaleontology **158**, 101890 (2020). https://doi.org/https://doi.org/10.1016/j.marmicro.2020.101890, https://www.sciencedirect.com/science/article/pii/S0377839819300830
3. Castillo, D.L.: Semtorch. https://github.com/WaterKnight1998/SemTorch (1988), [Online; accessed 28-August-2021]
4. Danescu, R., Pantilie, C., Oniga, F., Nedevschi, S.: Particle grid tracking system stereovision based obstacle perception in driving environments. Intelligent Transportation Systems Magazine, IEEE **4**(1), 6–20 (Spring 2012). https://doi.org/10.1109/MITS.2011.2178492
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
6. Howard, J.: Deep Learning 2019 - Fastai Course (2019), youTube, Jan. 25 [Video file], Available: https://youtu.be/XfoYk_Z5AkI . [Accessed: May. 14, 2020]
7. Howard, J., Gugger, S.: Fastai python library v1.0.57. http://docs.fast.ai/3 (2019), [Online; accessed 3-March-2020]
8. Hrnet. https://github.com/HRNet, [Online; accessed 29-August-2021]
9. Hrnet semantic segmentation. https://github.com/HRNet/HRNet-Semantic-Segmentation (1988), [Online; accessed 28-August-2021]
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014), http://arxiv.org/abs/1412.6980, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015
11. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A., van Ginneken, B., Sánchez, C.I.: A survey on deep learning in medical image analysis. Medical Image Analysis **42**, 60–88 (Dec 2017). https://doi.org/10.1016/j.media.2017.07.005, https://doi.org/10.1016/j.media.2017.07.005
12. Matias, A.V., Cerentini, A., Macarini, L.A.B., Amorim, J.G.A., Daltoé, F.P., von Wangenheim, A.: Segmentation, detection, and classification of cell nuclei on oral cytology samples stained with papanicolaou. SN Computer Science **2**(4), 285 (May 2021). https://doi.org/10.1007/s42979-021-00676-8, https://doi.org/10.1007/s42979-021-00676-8

13. Rateke, T., Justen, K.A., Chiarella, V.F., Sobieranski, A.C., Comunello, E., Wangenheim, A.V.: Passive vision region-based road detection: A literature review. ACM Comput. Surv. **52**(2), 31:1–31:34 (Mar 2019), doi: http://doi.acm.org/10.1145/3311951
14. Rateke, T., von Wangenheim, A.: Systematic literature review for passive vision road obstacle detection. Tech. rep., Brazilian Institute for Digital Convergence - INCoD (06 2018). https://doi.org/10.13140/RG.2.2.10198.14408
15. Rateke, T., von Wangenheim, A.: Passive vision road obstacle detection: a literature mapping. International Journal of Computers and Applications **0**(0), 1–20 (2020). https://doi.org/10.1080/1206212X.2020.1758877, https://doi.org/10.1080/1206212X.2020.1758877
16. Rateke, T., von Wangenheim, A.: Road obstacles positional and dynamic features extraction combining object detection, stereo disparity maps and optical flow data. CoRR **abs/2006.14011** (2020), https://arxiv.org/abs/2006.14011
17. Rateke, T., von Wangenheim, A.: Road surface detection and differentiation considering surface damages. Autonomous Robots **45**(2), 299–312 (Feb 2021). https://doi.org/10.1007/s10514-020-09964-3, https://doi.org/10.1007/s10514-020-09964-3
18. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR **abs/1505.04597** (2015), http://arxiv.org/abs/1505.04597
19. Smith, L.N.: No more pesky learning rate guessing games. CoRR **abs/1506.01186** (2015), http://arxiv.org/abs/1506.01186
20. Smith, L.N.: A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay. CoRR **abs/1803.09820** (2018), http://arxiv.org/abs/1803.09820
21. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514 (2019)
22. team, T.N.J.: Jetson tx2 developer kit user guide. https://developer.download.nvidia.com/embedded/L4T/r32-3-1_Release_v1.0/jetson_tx2_developer_kit_user_guide.pdf (2019), [Online; accessed 28-August-2021]
23. Victória Matias, A., Cerentini, A., Buschetto Macarini, L.A., Atkinson Amorim, J.G., Perozzo Daltoé, F., von Wangenheim, A.: Segmentation, detection and classification of cell nuclei on oral cytology samples stained with papanicolaou. In: 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS). pp. 53–58 (2020). https://doi.org/10.1109/CBMS49503.2020.00018
24. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2020). https://doi.org/10.1109/TPAMI.2020.2983686
25. Wang, S., Yang, D.M., Rong, R., Zhan, X., Xiao, G.: Pathology image analysis using segmentation deep learning algorithms. The American Journal of Pathology **189**(9), 1686–1698 (Sep 2019). https://doi.org/10.1016/j.ajpath.2019.05.007, https://doi.org/10.1016/j.ajpath.2019.05.007
26. von Wangenheim, A.: Fast pavement location (unpaved dirt road). [YouTube video] (Apr 26 2012), https://www.youtube.com/watch?v=jWMAbZKi7qY, accessed Sep. 1st, 2018
27. Won, K., Jung, S.: Billboard sweep stereo for obstacle detection in road scenes. Electronics Letters **48**(24), 1528–1530 (2012). https://doi.org/10.1049/el.2012.2353

28. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. pp. 173–190. Springer (2020)