Lightweight Uncertainty Quantification with Simplex Semantic Segmentation for Terrain Traversability

Judith Dijk[1], Gertjan J. Burghouts[1], Kapil D. Katyal[2], Bryanna Y. Yeh[3], Craig T. Knuth[3], Ella Fokkinga[1], Tejaswi Kasarla[4], Pascal Mettes[4]

***Abstract*— For navigation of robots, image segmentation is an important component to determining a terrain's traversability. For safe and efficient navigation, it is key to assess the uncertainty of the predicted segments. Current uncertainty estimation methods are limited to a specific choice of model architecture, are costly in terms of training time, require large memory for inference (ensembles), or involve complex model architectures (energy-based, hyperbolic, masking). In this paper, we propose a simple, light-weight module that can be connected to any pretrained image segmentation model, regardless of its architecture, with marginal additional computation cost because it reuses the model's backbone. Our module is based on maximum separation of the segmentation classes by respective prototype vectors. This optimizes the probability that out-of-distribution segments are projected in between the prototype vectors. The uncertainty value in the classification label is obtained from the distance to the nearest prototype. We demonstrate the effectiveness of our module for terrain segmentation.**

I. Introduction

Safe and efficient robot off-road navigation highly depends on accurate and actionable information of the environment. Semantic segmentation is a common component to determining the terrain's traversability with images (e.g. [1]). This segmentation provides information about the current environment such as types of surfaces (e.g. puddles vs. dirt) or obstacles (e.g. tall grass vs. tree) that could impact robot navigation. If this segmentation model can run (almost) real-time on the robot, it can be used for path planning or replanning.

Not only the labels themselves are important, but also an estimate of their uncertainty. Such an uncertainty estimation enables online reasoning in path planning, e.g. uncertain areas can be avoided or entered more carefully. Such an approach is e.g. shown by Cai et al. [2] where traversability estimates obtained by traction parameters of the platform are used for risk estimation, and by Hakobyan et al. [3] who proposed risk-aware motion planning and control Using conditional value-at-risk (CVaR)-Constrained Optimization.

Standard methods for semantic segmentation focus on label accuracy and not on the accuracy of the uncertainty estimate. Segmentation models that provide uncertainty are not optimized for robotics and embedded scenarios [4], [5], where uncertainty quantification needs to be fast, compact, and without impacting the segmentation accuracy.

The approach presented in this paper consist of a simple, lightweight module for uncertainty estimation for image segmentation. This module, dubbed Simplex Semantic Segmentation, can be connected to any pretrained semantic segmentation model as it is architecture agnostic. Our approach is based on the prototypes approach that was developed for classification, often on imbalanced datasets with rare classes [6]–[11]. In this paper, we apply this prototype approach to obtain the uncertainty for semantic segmentation, where each *pixel* of an input image needs to be labeled.

This paper is organized as follows: Section II presents our method for the prototype module and the uncertainty estimation. In Section III, these proposed methods are evaluated. In Section IV, the findings are summarized and discussed.

II. Proposed approach

A. Rationale

To estimate the uncertainty of image segmentation, we propose a simple, light-weight module that can be connected to any pretrained segmentation model. The module only needs a feature map and is independent of the underlying architecture. As the module is only an extra model head, it only adds a marginal computation cost. Our module builds on the prototypes approach, that is commonly used to improve classification of imbalanced datasets with rare classes [6]–[8]. In the standard prototype approach, each segmentation class is represented by its respective prototype vector. In our proposed method, we apply this prototype approach to obtain the uncertainty for semantic segmentation, where each *pixel* of a given input image is labeled to a specific class. In the proposed method, we apply this prototype approach to obtain the *uncertainty* the labeled pixel. During training, these prototypes are incorporated to maximize distance between classes. In the inference phase, samples are then classified based on their distance to prototypes. The prototype approach maximizes the probability that pixels of unknown image segments will be projected in the void space between the unknown classes. The uncertainty is inferred from the distance to the nearest prototype vector. Figure 1 provides an overview of our approach and how the proposed module is connected to any segmentation model.

*This work was partly supported by TNO Appl.AI program

[1] Judith Dijk, Gertjan Burghouts and Ella Fokkinga are with the department of Intelligent Imaging, TNO, 2597 AK, the Netherlands judith.dijk@tno.nl

[2]Kapil Katyal is with Johns Hopkins University, Baltimore, MD, USA

[3]Bryanna Yeh and Craig Knuth are with the Johns Hopkins University Applied Physics Laboratory, Laurel, MD, USA bryanna.yeh@jhuapl.edu

[4]Tejaswi Kasarla and Pascal Mettes are with the University of Amsterdam, the Netherlands.

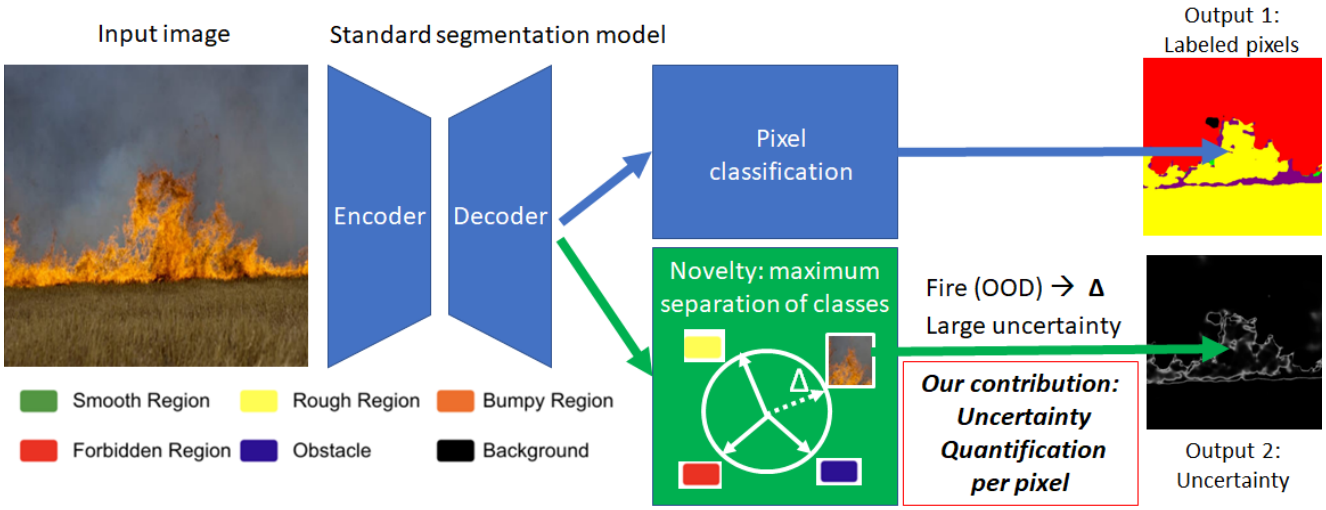


Fig. 1: This figure describes our ability to predict segmentation classes with uncertainty. The input image consists of regions that are in-distribution and areas that are out-of-distribution. We are able to extend a traditional segmentation model (a) by estimating uncertainty through distances to the nearest prototype vector (b). In this example, we are able to detect out-of-distribution areas (fire) through our uncertainty estimation.

B. Method

The starting point of our approach is a given pretrained image segmentation model $h\colon \mathcal{X} \rightarrow \mathcal{Y}$ which classifies image pixels x_n into N respective classes y_n using a set of classification parameters W_c. We rewrite $h(\cdot; W_c) = g(f(\cdot; W_f); W_g)$, where $f(\cdot; W_f)$ is the model's backbone which outputs a feature map, and $g(\cdot; W_g)$ is the pixel classification head that transforms the feature map into class predictions. W_f and W_g are the parameters connected to the functions f and g, respectively. Our objective is to predict the uncertainty of the labels using parameters W_u by the module $U = u(f(\cdot; W_f); W_u)$, with W_u the uncertainty estimation parameters. Note that this uncertainty function U can be applied to any spatial feature map f, without loss of generality, which makes our approach applicable to any segmentation model.

The uncertainty function U can be further separated using prototype vectors P

$$U = m(l(\cdot, W_u) \cdot P) \tag{1}$$

where m is a fixed mapping and $l(\cdot, W_u)$ a neural network that projects pixel features onto the respective class prototype to maximize the output value for the pixel's correct segmentation class [8].

The prototype vectors are maximally separated on the $(N-1)$ dimensional hypersphere. The derivation is recursive [12]:

$$P_1 = \begin{pmatrix} 1 & -1 \end{pmatrix} \in \mathbb{R}^{1\times 2} \tag{2}$$

$$P_k = \begin{pmatrix} 1 & -\frac{1}{k}\mathbf{1}^T \\ \mathbf{0} & \sqrt{1-\frac{1}{k^2}}\, P_{k-1} \end{pmatrix} \in \mathbb{R}^{k\times(k+1)} \tag{3}$$

with $\mathbf{0}$ and $\mathbf{1}$ respectively the 0 and 1 column vectors. The columns of P_k are $k+1$ equidistant vectors on the unit sphere in $\mathbb{R}^k$. With N classes, the prototypes are obtained by constructing P_{N-1}, yielding N vectors in $N-1$ dimensions. The rationale is that if the pixel's projection is further away from the prototype vector, it is more uncertain.

The uncertainty is inversely proportional to the maximum of a pixel's scores across the class prototypes: $m(\mathbf{x}) = 1 - \sigma(max(\mathbf{x}))$, where $\mathbf{x}$ are the pixel's prototype scores from $l(\cdot, W_u) \cdot P$, and σ is the softmax operator. Note that this uncertainty is not calibrated yet.

To predict uncertainty, the module needs to learn the classes first. Therefore, the learning objective is to align the outputs of $l(f(x_i), W_u)$ with the class prototypes P_{y_i}, with x_i and y_i from the training set $\mathcal{S}$, using the cosine loss:

$$\mathcal{L} = -\sum_{i=1}^{M} \frac{m(l(f(\mathbf{x}_i), W_u)) \cdot P_{y_i}}{\|m(l(f(\mathbf{x}_i), W_u))\| \cdot \|P_{y_i}\|} \tag{4}$$

After learning to project a pixel's feature onto the prototype vector, at inference we can tell if a test pixel is projected far away from such a vector. The further away, the more uncertain the pixel's classification. We hypothesize that this works best if the prototype vectors are maximally apart.

III. Experiments

A. Base dataset

Since we target off-road applications, we use a base dataset recorded in off-road environments with a diverse range of objects and terrain types. From the online available datasets [13]–[15] we choose Rellis3D [14], which is collected in an off-road environment with 6,235 annotated images. The classes in this dataset are Concrete, Asphalt, Gravel, Grass, Dirt, Sand, Rock, Rock Bed, Water, Bushes, Tall Vegetation, Trees, Poles, Logs, Void, Sky, Sign. The environment is complex and differs between images. To segment the images in terms of traversability, we adopt the

six classes from [16]: Smooth, Rough, Bumpy, Forbidden, Obstacles and Background.

B. Model training

For the experiments, we select DeepLabV3+ [17] for its simplicity and broad usage. DeepLabv3+ contains a decoder that refines the segmentation using atrous convolutions and a spatial pyramid pooling. As a backbone, we select a Resnet50 [18] pretrained on ImageNet [19], as this is one of the most commonly used backbones. The images are resized to 512 $\times$ 512 pixels and augmented with standard transformations: horizontal flip, shift, scale, rotate and color jitter (all at a probability of 0.5). The batch size is 4. The model is trained for 25 epochs, at a learning rate of 0.001. All weights, including the backbone, are optimized during training. The segmentation results and accuracy of the segmentation are presented in the Appendix.

C. Uncertainty estimation

For the uncertainty estimation, we start by comparing the uncertainty values measured in the test set of Rellis3D and the uncertainty values measures in images from other datasets. The uncertainty values should be higher for datasets that are different. For simplicity, we assume that all pixels in an image are certain for Rellis3D and uncertain for other datasets, because for these datasets we do not have detailed annotations of certain or uncertain image segments. This assumption is often violated, because images from the other datasets may have segments that are very similar to Rellis3D. We compare against the DeepLabV3+ baseline, which is the same model, but without our uncertainty module. Its outputs are transformed into uncertainty values by the mapping $m(\cdot)$ from Equation 4. This methods is referred to a *standard method* in the remainder of this paper.

The uncertainty performance is measured by the 1) Receiver Operator Characteristic (ROC) curve (graph), in which the true positive rate is plotted against the False Positive Rate and the 2) Area Under the Curve (AUC). The best performance will be found if the left upper corner is reached, in which case the AUC is 1.

The AUC scores for both methods for all datasets is shown in Table I. The ROC curves can be seen in Figure 6 in the Appendix. On all datasets, our method shows a better uncertainty estimation performance than the standard method. On CUB-200, MS-COCO and KITTI, the performance is very similar with the standard method. These datasets are reasonably different from Rellis3D, but often contain vegetation, streets, humans and cars, which are also in Rellis3D. For WiderPerson and Fukuoka, our method is favorable, which can be explained by respectively the different viewpoint (aerial) and environment (indoor). The most prominent insight is that our method is uncertain about fog and fire, whereas the standard method is as certain as it is for Rellis3D on which it was trained. This result demonstrates the merit of our method for uncertainty estimation in practical cases.

TABLE I: AUC for the standard method and the proposed method. The datasets are ordered from low to high difference with Rellis3D.

Dataset	Contents	standard method	our method
MS-COCO [20]	common objects	0.766	0.794
WiderPerson [21]	aerial	0.796	0.846
KITTI [22]	self-driving cars	0.853	0.859
SceneParse150 [23]	common segments	0.770	0.861
Fukuoka [24]	indoor robot	0.847	0.904
CUB-200 [25]	birds	0.841	0.856
Fog and Fire	outdoor anomalies	0.478	0.881

D. Segment-specific uncertainty

We evaluate the uncertainty estimation for respective segment classes across images. In the SceneParse150 [23] dataset, most segmentation classes are semantically and visually different from the Rellis3D classes. The performance is measured by the AUC on uncertainty values per segment class in comparison to Rellis3D uncertainty values. The segmentation classes that are most uncertain are shown in Figure 2. Indeed, these classes are most different from Rellis3D, hence are expected to yield high uncertainties. In comparison to the standard method, our method yields higher uncertainty values for the segmentation classes that deviate from the training dataset.

E. Uncertainty visualization

It is also possible to show the uncertainty per pixel. An example is shown in Figure 1. Here the *Input image* shows both in-domain (vegetation, sky) and out-of-domain (fire) regions. *Output1* shows the segmentation and *Output2* shows the resulting uncertainty values: low (black) for the in-domain image segments and high (white) for the out-of-domain image segments. This is the desired behaviour: the robot can make navigation decisions about the image segments where it is certain, while operating in safe mode at the uncertain segments. It can be seen that the more uncertain values are found around the edges of segments, which is as expected as these pixels might contain information from both of these segments, and the labeling will often be based on surrounding pixels. This can also be seen in Figure 3 for two images from the SceneParxe10 dataset, where the uncertainty found for the unknown class of the pigs is much higher with our proposed method than with the standard method. More examples are presented in Figure 5 in the Appendix. In the bottom image, it can also be seen that the uncertainty for the field is much smaller for our method than for the standard method. This smaller uncertainty will allow for better path planning and faster navigation, by avoiding paths with high uncertainty.

F. Inference computational costs

Regarding the computational cost of inference on a test image, our method is advantageous. The most common method for uncertainty evaluation is to use Monte-Carlo dropout [26]. Although very effective, this requires a number of repeated feed-forward calculations of the model with

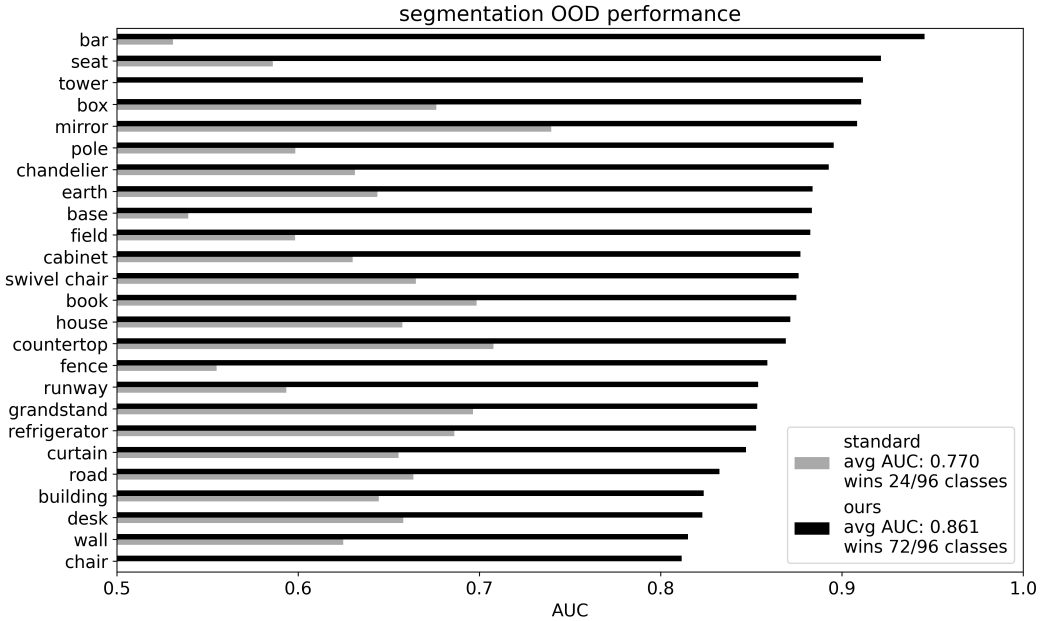


Fig. 2: Segments from SceneParse150 for which our model (trained on Rellis3D) assigns the highest uncertainties.

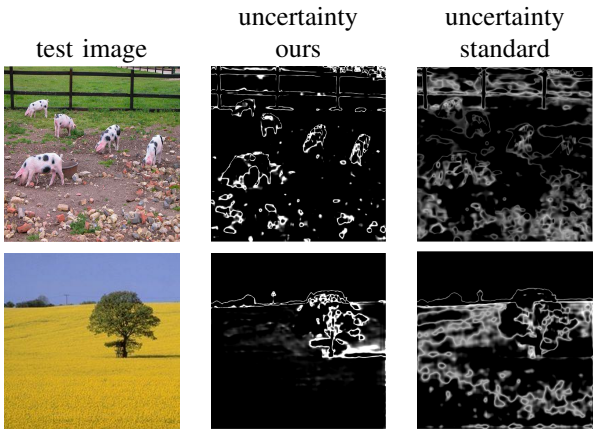


Fig. 3: Our model (middle) yields high uncertainty values for the tree, pigs and fence, whereas the uncertainty obtained with the standard method is scattered (right).

randomly sampled weight parameters, which needs much computation power and can cause long latency. In our experiments, the model is DeepLabV3+ which has 11.9M weights. Instead of running this model multiple times, our module is low-cost and has to be run only once. For six traversability classes, our module has 1285 weights, i.e. 0.01% of the total model. These weights are required to transform the model's feature map (the final upscaling layer), a tensor of 512 $\times$ 512 $\times$ 256 dimensions, into 512 $\times$ 512 $\times$ 5 dimensions. This transformation is $l(\cdot, W_u)$ in Equation 1. It is implemented as a channel-wise convolution from 256 to 5 dimensions. The mapping P from Equation 1 projects the output of $l(\cdot, W_u)$ from 5 to 6 dimensions, which are the 6 traversability classes. This projection is a fixed matrix multiplication, no learnable weights, implemented as a highly efficient vectorized operation.

IV. Conclusions and discussion

We propose an extra, lightweight module for a semantic segmentation network, which provides a high quality segmentation label per pixel and an uncertainty estimate for these labels. The module first optimizes the segmentation by maximizing the separation of the different classes in the training phase. In the inference phase, the uncertainty of a pixel can be estimated as the distance of the current pixel to the center of the class it is labeled to. We have shown that our approach performs on par (with GA-Nav-r8) or just a little (for DeepLabV3++) better than standard models. We have shown high uncertainty values for data that really differs from the Rellis3D data we trained on. This indicates that the added module provides a good uncertainty estimate for pixels and segments in an image.

In future work, We will evaluate the uncertainty estimation in a more quantitive way by estimating the 'ground truth uncertainty' using monte carlo estimations. Next to that, we will compare our approach against existing uncertainty estimation approaches. We will also focus on on calibrating the uncertainty. The uncertainty visualisation shows that our labeling is more uncertain on edges in the image. When using the labeling and uncertainty estimation for navigation, this means that these areas can be avoided when possible. In future work we will implement this simplex semantic segmentation and uncertainty approach for robust long-distance navigation on a physical robot, both for providing trusted generalized features for self-supervised traversability prediction as in [27] and informing risk-based planning such as CVaR-Conditional optimization mentioned earlier. This will provide information on how well this uncertainty can actually be used for navigation purposes.

REFERENCES

[1] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds. Cham: Springer International Publishing, 2018, pp. 335–350.

[2] X. Cai, M. Everett, L. Sharma, P. R. Osteen, and J. P. How, "Probabilistic traversability model for risk-aware motion planning in off-road environments," *arXiv preprint arXiv:2210.00153*, 2022.

[3] A. Hakobyan, G. C. Kim, and I. Yang, "Risk-aware motion planning and control using cvar-constrained optimization," *IEEE Robotics and Automation letters*, vol. 4, no. 4, pp. 3924–3931, 2019.

[4] M. G. Atigh, J. Schoep, E. Acar, N. Van Noord, and P. Mettes, "Hyperbolic image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4453–4462.

[5] A. Jungo and M. Reyes, "Assessing reliability and challenges of uncertainty estimations for medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention*. Springer, 2019, pp. 48–56.

[6] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[7] J. Cen, P. Yun, J. Cai, M. Y. Wang, and M. Liu, "Deep metric learning for open world semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 333–15 342.

[8] T. Kasarla, G. Burghouts, M. van Spengler, E. van der Pol, R. Cucchiara, and P. Mettes, "Maximum class separation as inductive bias in one matrix," *Advances in Neural Information Processing Systems*, vol. 35, pp. 19 553–19 566, 2022.

[9] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song, "Learning towards minimum hyperspherical energy," *Advances in neural information processing systems*, vol. 31, 2018.

[10] P. Mettes, E. Van der Pol, and C. Snoek, "Hyperspherical prototype networks," *Advances in neural information processing systems*, vol. 32, 2019.

[11] J. Zhou, C. You, X. Li, K. Liu, S. Liu, Q. Qu, and Z. Zhu, "Are all losses created equal: A neural collapse perspective," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31 697–31 710, 2022.

[12] user1551, "How to find $n+1$ equidistant vectors on an n-sphere?" Mathematics Stack Exchange, version: 2018-03-07. [Online]. Available: https://math.stackexchange.com/q/714781

[13] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5000–5007.

[14] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 1110–1116.

[15] O. Mayuku, B. W. Surgenor, and J. A. Marshall, "Multi-resolution and multi-domain analysis of off-road datasets for autonomous driving," in *2021 18th Conference on Robots and Vision (CRV)*. IEEE, 2021, pp. 165–172.

[16] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "GA-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.

[17] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.

[21] S. Zhang, Y. Xie, J. Wan, H. Xia, S. Z. Li, and G. Guo, "Widerperson: A diverse dataset for dense pedestrian detection in the wild," *IEEE Transactions on Multimedia*, vol. 22, no. 2, pp. 380–393, 2019.

[22] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307.

[23] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[24] O. Martínez Mozos, K. Nakashima, H. Jung, Y. Iwashita, and R. Kurazume, "Fukuoka datasets for place categorization," *The International Journal of Robotics Research*, vol. 38, no. 5, pp. 507–517, 2019.

[25] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," 2011.

[26] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059.

[27] A. Polevoy, C. Knuth, K. M. Popek, and K. D. Katyal, "Complex terrain navigation via model error prediction," *CoRR*, vol. abs/2111.09768, 2021. [Online]. Available: https://arxiv.org/abs/2111.09768

APPENDIX: ILLUSTRATIONS

In this appendix, we show some figures illustrating the results presented earlier.

In Figure 4 the segmentation results on test images from the Rellis3D dataset are shown. These cases are in-domain. When comparing the results to the ground-truth labels, it can be seen they are mostly correct. The uncertainty estimation for our method and the standard method are also shown. It can be seen that larger uncertainty values are assigned to unclear boundaries between segments for the standard method (last than for our method.

In Figure 5 The uncertainty estimation for our method and the standard method are shown for images with out-of-domain regions: for two images from our own dataset, and four images from the SceneParse150 [23] dataset.The uncertainty estimates with our method are higher than those obtained with the standard method.

In Figure 6 the ROC curves for six datasets are shown. These graphs show that our method shows a better uncertainty estimation performance than standard method.

A. Segmentation results and accuracy

To evaluate the segmentation results of our proposed method, we compare the results with two state-of-the-art methods: DeepLabV3+ as our method is added on top of this method and GA-Nav [16]. We report the Intersection over Union (IoU) for each class i and the mean IoU (mIoU).

Table II shows that our method performs on par with GA-Nav-r8. Surprisingly, our model performs better than the reported DeepLabV3+ in [16], which has the same architecture and setup as model, but was trained with other settings. We hypothesize that some of our design choices are better suited for Rellis3D. One of the differences with [16] is that they train with a smaller batch size (2 instead of 4). Their image size is 375 $\times$ 600 pixels, which corresponds less with the aspect ratio of the images in the dataset (1600 $\times$ 1920

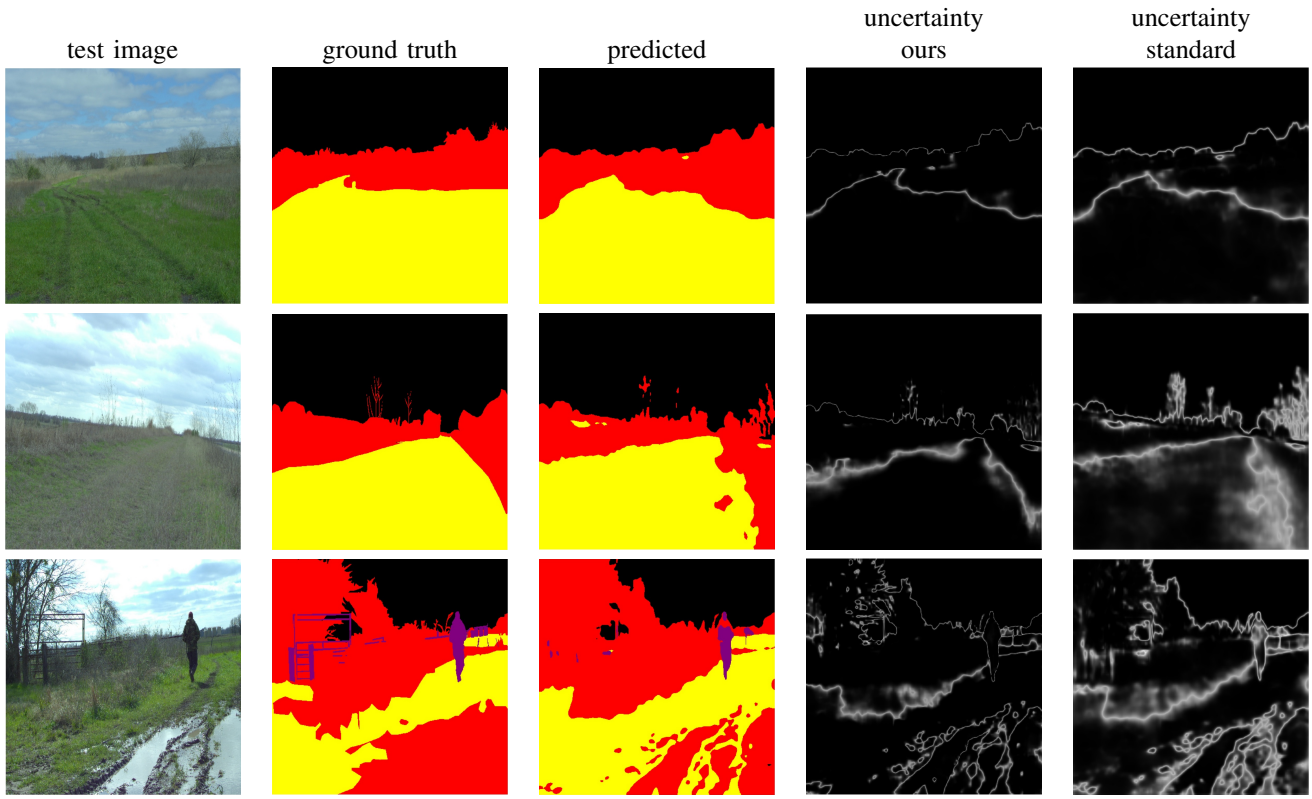


Fig. 4: In-domain cases from the Rellis3D dataset. The segmentation predictions are often correct (third column) and larger uncertainty values are assigned to unclear boundaries between segments for the standard method (last column) than our method (fourth column).

TABLE II: Segmentation results for the Rellis3D dataset [14] in mIoU for all classes for the DeepLabV3+ [16], [17] model, the GA-Nav-r8 [16] model and our approach.

Model	smooth	rough	bumpy	forbidden	obstacle	background	mean
DeepLabV3+	65.8	79.8	19.7	47.5	64.9	95.9	62.3
GA-Nav-r8	78.5	88.3	37.3	72.3	74.8	96.1	74.4
Our method	84.7	86.8	59.9	80.2	43.5	97.0	**75.3**

pixels) than our image size (512 $\times$ 512 pixels). Possibly the biggest advantage is that we use more augmentations. Where [16] uses only horizontal flip and random crop, we add scale, rotate and color jitter. Especially the scale changes in the dataset are sometimes large, which we address by the scale augmentation. Some of the images are recorded under a slight tilt, because the robot might be on a slope. This is addressed with rotation augmentations during training. Examples of the segmentation images are shown in the Appendix in Figure 4.

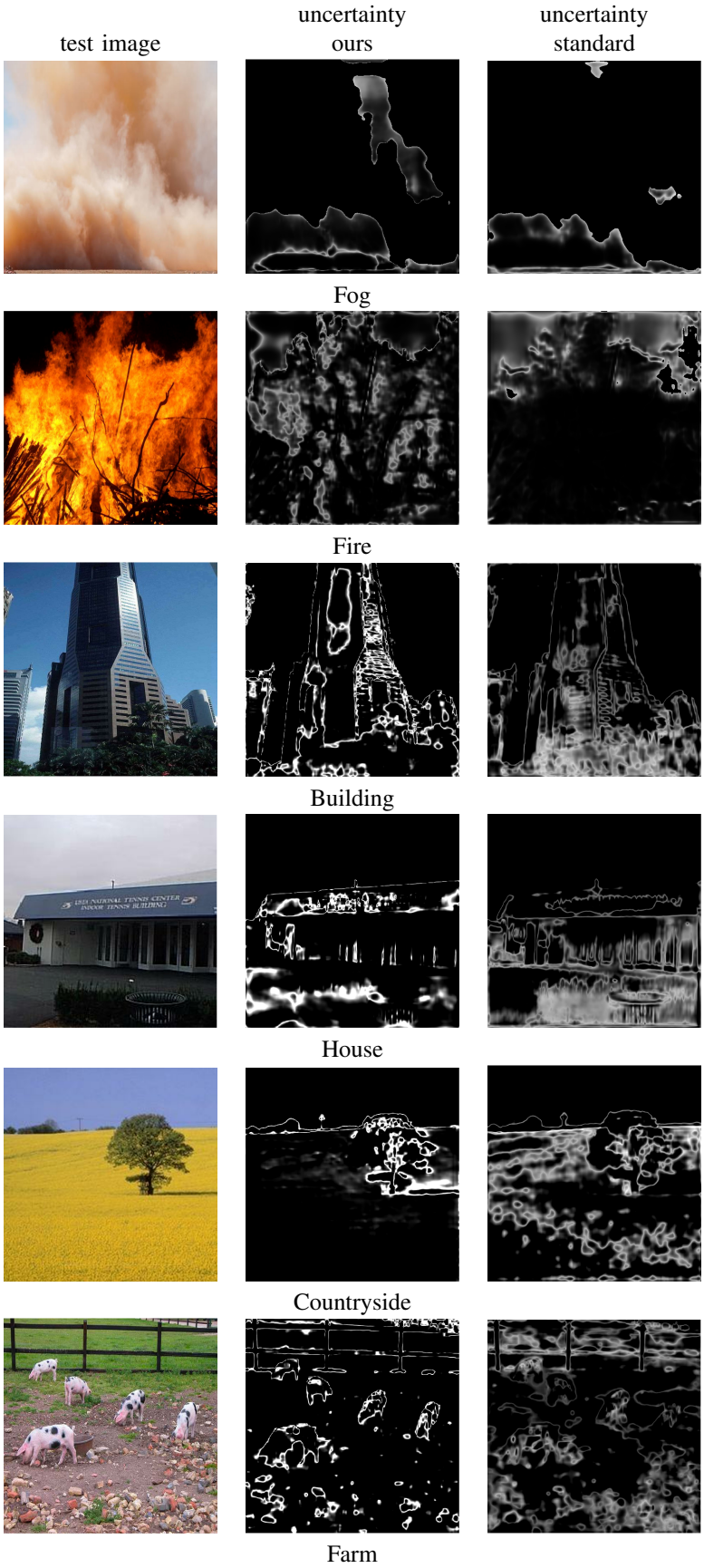
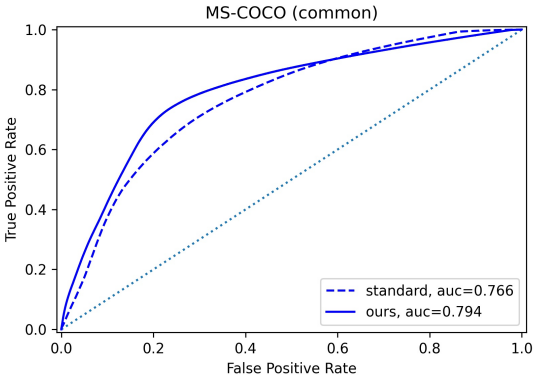
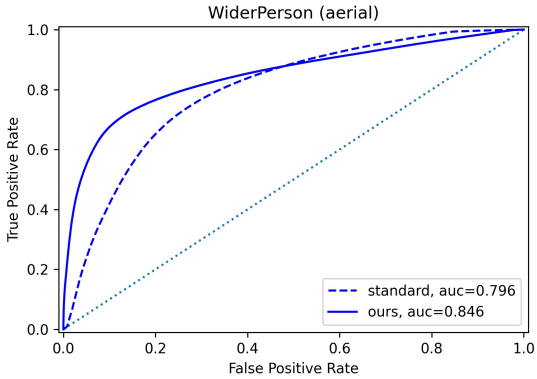


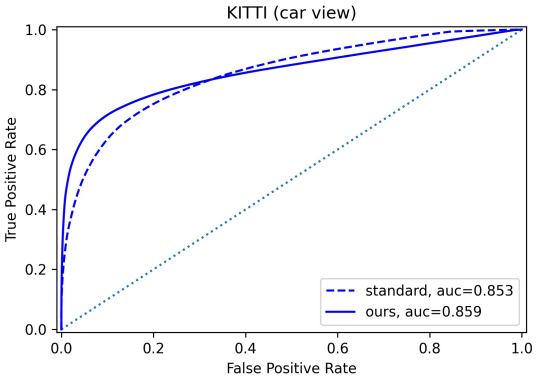
Fig. 5: Visualisation of the uncertainty for images with out-of-domain regions: two images from our dataset (fog and fire) and four images from the SceneParse150 [23] dataset (building, house and countryside).



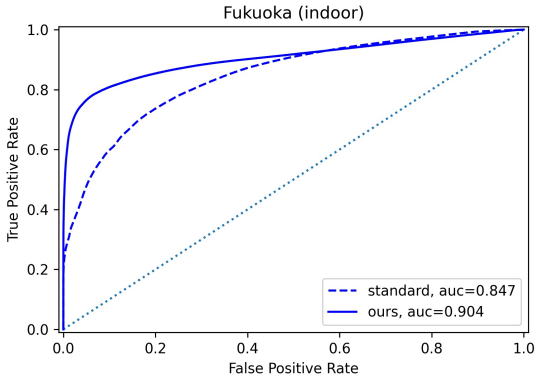
MS-COCO: common objects



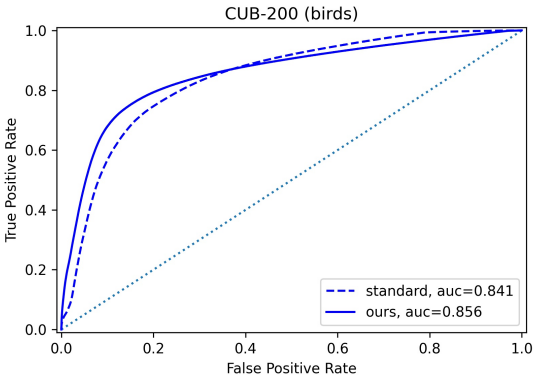
WiderPerson: aerial



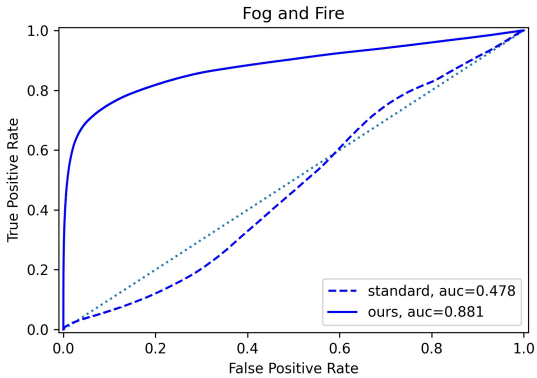
KITTI: self-driving cars



Fukuoka: indoor robot



CUB-200: birds



Fog and fire: anomalies

Fig. 6: Out-of-distribution performance of our uncertainty estimation on various datasets. The AUC values are also reported in Table I