Out-of-Distribution Detection Using Deep Neural Network Latent Space Uncertainty

Fabio Arnez[1],*, Ansgar Radermacher[1] and François Terrier[1]

[1]Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

Abstract

As automated systems increasingly incorporate deep neural networks (DNNs) to perform safety-critical tasks, confidence representation and uncertainty estimation in DNN predictions have become useful and essential to represent DNN ignorance. Predictive uncertainty has often been used to identify samples that can lead to wrong predictions with high confidence, i.e., Out-of-Distribution (OoD) detection. However, predictive uncertainty estimation at the output of a DNN might fail for OoD detection in computer vision tasks such as semantic segmentation due to the lack of information about semantic structures and contexts. We propose using the DNN uncertainty from intermediate latent representations to overcome this problem. Our experiments show promising results in OoD detection for the semantic segmentation task.

Keywords

Uncertainty Estimation Latent Space Out-of-Distribution Detection Semantic Segmentation Automated Vehicle

1. Introduction

In the last decade, Deep Neural Networks (DNNs) have witnessed great advances in real-world applications like Autonomous Vehicles (AVs) to perform complex tasks such as object detection and tracking or vehicle control. Despite the progress introduced by DNNs in the previous decade, they still have significant safety shortcomings due to their complexity, opacity and lack of interpretability. Moreover, it is well-known that DNN models behave unpredictably under dataset shift [1]. Deep Learning (DL) models have training and data bias that directly impact model predictions and performance. This impedes ensuring the reliability of the DNN models, which is a precondition for safety-critical systems to ensure compliance with industry safety standards to avoid jeopardizing human lives [2].

As highly automated systems (e.g., autonomous vehicles or autonomous mobile robots) increasingly rely on DNNs to perform safety-critical tasks, different methods have been proposed to represent confidence in the DNN predictions. One way to represent DNN confidence is to capture the uncertainty associated with a prediction for a given input sample. Capturing information about *"what the model does not know"* is not only useful but essential in safety-critical tasks.

Bayesian Neural Networks (BNNs) and existing Bayesian approximate inference methods (Deep Ensembles, Monte-Carlo Dropout, etc.) offer a principled approach to model and quantify uncertainties in DNNs. However, quantifying uncertainty is challenging since we do not have access to ground-truth uncertainty estimates, i.e., we do not have a clear definition of what a good uncertainty estimate is. Moreover, computer vision tasks can add an extra level of complexity since tasks such as semantic segmentation require a pixel-level understanding of an image. In this case, a Bayesian Deep Learning model for semantic segmentation will classify each pixel in the input image and generate an uncertainty estimate for each classified pixel.

In semantic segmentation, uncertainty estimation has been used for Out-of-Distribution (OoD) detection under the assumption that samples that are far away from the training distribution (anomalous or OoD samples) provide higher predictive uncertainty than samples that are observed in the training data [3]. Approaches that use BNNs are able to capture aleatoric and epistemic uncertainties in the form of uncertainty maps (Figure 1-top) but still fail to detect anomalies accurately. BNN methods for semantic segmentation are prone to yield false-positive predictions, as well as miss-matches between anomaly instances and uncertain areas caused by the lack of information on semantic structures and contexts [4, 5], as presented in Figure 1-middle.

Recently, embedding density estimation methods have been proposed to estimate the connection to uncertainties from Bayesian methods [6, 3]. In this direction, methods that leverage metrics or statistics from the non-parametric embedding space density have been proposed recently [7, 8], in contrast to a distance-based method that often assumes a parametric embedding density [9, 10, 11].

The present work combines the benefits from Bayesian methods for uncertainty estimation with methods for la-

The 37th AAAI Conference on Artificial Intelligence: SafeAI 2023 workshop, February 07–14, 2023, Washington, DC, USA

*Corresponding author.

fabio.arnez@cea.fr (F. Arnez); ansgar.radermacher@cea.fr (A. Radermacher); francois.terrier@cea.fr (F. Terrier)

0000-0003-0367-3035 (F. Arnez)

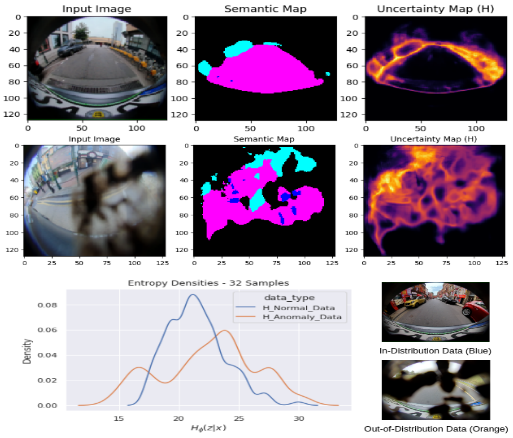

Figure 1: Semantic segmentation uncertainty estimation comparison for in-distribution and out-of-distribution data

tent representation density estimation in the OoD detection task. We propose to capture the entropy of intermediate (latent) representations and estimate the entropy densities for In-Distribution (InD) and OoD samples (see Figure 1-bottom). Once entropy densities are estimated, we use them to classify new input samples as InD or OoD, i.e., we build a data-driven monitoring function data that utilizes the input sample entropy for the OoD detection task.

2. Semantic Segmentation with Probabilistic U-Net Architecture

Probabilistic U-Net [12], is a DNN architecture for semantic segmentation that combines the U-Net architecture [13] with the conditional variational autoencoder (CVAE) framework [14]. The goal of Probabilistic U-Net is to handle input image ambiguities by leveraging the stochastic nature of the CVAE latent space. Figure 2 shows the Probabilistic U-Net architecture.

During training, depicted in Figure 2a, Probabilistic U-Net finds a useful embedding of the segmentation variants in the latent space by introducing a Posterior Net. This network learns to recognize a segmentation variant and to map it into a noisy position in the latent space $(\mu_{post}, \sigma^2_{post})$. In addition, KL divergence is used to penalize differences between the distributions at the output of prior and posterior nets. The idea here is to bring both distributions as close as possible so that the Prior Net distribution covers the spaces of all presented segmentation variants.

In general, the central component of this architecture is its latent space. Each value from the latent space encodes a segmentation variant. During inference, the Prior Net encodes each input image x_i and estimates the probability of these segmentation variants $(\mu_{prior}, \sigma^2_{prior})$. To predict a set of segmentation outputs, a set of samples are drawn from the Prior Net probability distribution. Interestingly, we can draw a connection from this approach to other related work that aims to model complex aleatoric uncertainty (ambiguity, multi-modality) by handling stochastic input variables [15, 16, 17].

3. Methods

3.1. Capturing Uncertainty from Intermediate Latent Representations

Despite the benefits introduced by injecting random samples from the latent space into U-Net, aleatoric uncertainty alone is not enough. For the Out-of-Distribution detection task, epistemic uncertainty is needed [18, 19]. Although the Prior Net encoder q_{prior} employs Bayesian inference to obtain latent vectors $\mathbf{z}$, it does not capture epistemic uncertainty since the encoder lacks a distribution over parameters ϕ. To overcome this problem, we took inspiration from Daxberger and Hernández-Lobato [20], Jesson et al. [21], and propose to capture uncertainty in the Probabilistic U-Net Prior Net encoder using M Monte Carlo Dropout (MCD) samples [22], i.e., $q_{prior}(z \mid x, \phi_m)$.

$$q_\Phi(\mathbf{z} \mid \mathbf{x}, \mathcal{D}_p) = \int_\phi q(\mathbf{z} \mid \mathbf{x}, \phi) p(\phi \mid \mathcal{D}_p) d\phi \qquad (1)$$

In eq. 1, we adapt the Prior Net encoder to capture the posterior $q(\mathbf{z} \mid \mathbf{x}, \mathcal{D})$ using a set $\Phi = \{\phi_m\}_m^M$ of encoder parameters samples $\phi_m \sim p(\phi \mid \mathcal{D}_p)$ that are obtained applying MCD at test-time. During execution time, we forward-pass an input image x_i multiple times into the q_{prior} net. Each time we forward-pass the input image, we will generate a new dropout mask that in consequence will make a new $(\mu_{prior}, \sigma^2_{prior})$ prediction. From each predicted $(\mu_{prior}, \sigma^2_{prior})$ for the same image we sample a new latent vector $\mathbf{z}$, as presented in Figure 3.

MCD has been applied extensively for simple *epistemic* uncertainty estimation. However, dropout was found to be ineffective on convolutional neural networks (CNNs). Standard dropout is ineffective in removing semantic information from CNN feature maps because nearby activations contain closely related information. On the other hand, dropping continuous regions in 2D feature maps can help remove semantic information and enforce remaining units to learn features for the assigned task [23]. This effect is also desired for capturing uncertainties, otherwise, we could get overconfident uncertainty estimates in the presence of samples that contain anomalies. To overcome the standard dropout limitation, we followed

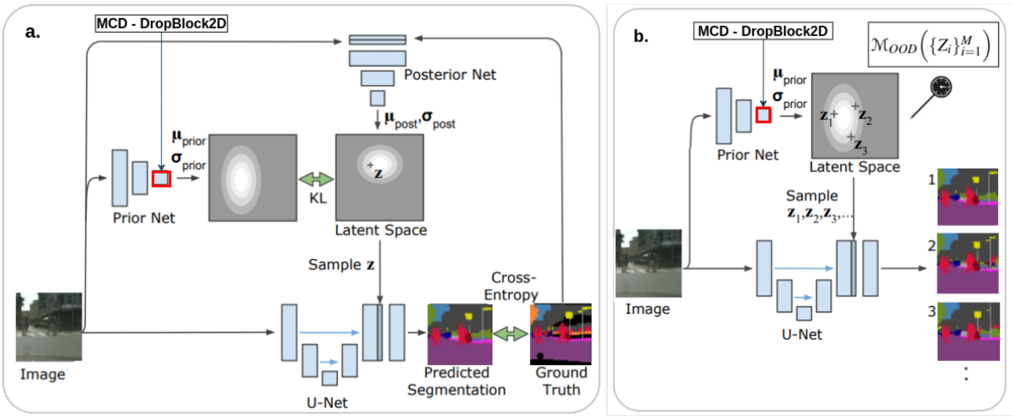

Figure 2: Probabilistic U-Net [12], with *Bayesian* Prior Net for Semantic Segmentation: **a.** During training **b.** During inference with the monitoring function $\mathcal{M}_{OOD}$ at the output of the Prior Net.

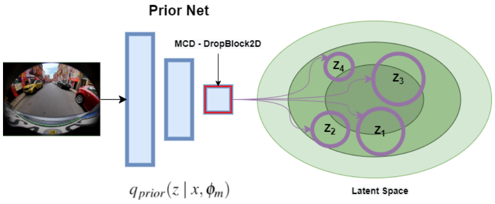

Figure 3: Prior Net latent vector $\mathbf{z}$ predictions with Monte Carlo DropBlock2D. The latent space at the output of the Prior Net is presented in 2D for illustration purposes.

the approach from Deepshikha et al. [24], and used DropBlock2D to capture uncertainty from the Probabilistic U-Net. We applied MC DropBlock2D in the last feature map from the Prior Net, as shown in Figure 2 and Figure 3 (in red).

The average surprise or uncertainty of a random variable z is defined by its probability distribution $p(z)$, and it is called the entropy of z, i.e., $\mathbb{H}(z)$. For continuous random variables, we use the differential entropy, as presented in Eq. 2,

$$\mathbb{H}(z) = \int_z p(z) \log \frac{1}{p(z)} dz \tag{2}$$

To quantify uncertainty from Prior Net MCD samples, we used standard entropy estimators [25] on 32 Monte Carlo samples (32 image forward passes through Prior Net with MC DropBlock2D turned on). In Eq. 3, the entropy $\hat{\mathbb{H}}_{\Phi}(z \mid x)$ measures the average surprise of observing latent vector z at the output of Prior Net, given an input image x.

$$\mathbb{H}(z \mid x) = \int_z p(z \mid x) \log \frac{1}{p(z \mid x)} dz \tag{3}$$

3.2. Bayesian Generative Classifier for OoD Detection

For OoD detection, we assume that we have access to a dataset of normal (InD) and anomaly (OoD) samples $Y = \{\text{normal}, \text{anomaly}\}$, with which we can train a Bayesian generative classifier (*Not so naive Bayes Classifier*) using the empirical density of a metric or statistic T from latent representations $\mathbf{z}$, i.e., $T(\mathbf{z})$. To this end, we follow Morningstar et al. [7] approach and use a Kernel Density Estimation (KDE) method to obtain the $T(\mathbf{z})$ densities. Since we aim at leveraging the uncertainty from intermediate latent representations, the T statistic is the entropy at the output of the Prior Net (described in the previous section) with which we build the monitoring function $\mathcal{M}_{OOD}$, as presented in Figure 2b.

For each label set, we fit a KDE to obtain a generative model of the data, i.e., use KDE to compute the likelihood $p(T(\mathbf{z}) \mid y)$. Then, we compute the class label prior probability $p(Y)$, i.e., compute the marginal categorical distribution by counting frequencies (from the number of samples of each class in the complete training set). For an unknown latent vector, we can compute the posterior probability of each class $p(y \mid T(\mathbf{z}))$, using Baye's rule in Eq. 4. For the OoD task, we use Eq. 5

$$p(y \mid T(\mathbf{z})) = \frac{P(T(\mathbf{z}) \mid y)p(y)}{p(T(\mathbf{z}))} \tag{4}$$

$$p(y \mid T(\mathbf{z})) = \frac{p(T(\mathbf{z}) \mid y)p(y)}{\sum_{y \in Y} p(T(\mathbf{z}) \mid y)p(y)} \tag{5}$$

For a more details description of the approach for Bayesian generative classification we refer the reader to the works from VanderPlas [26] and Postels et al. [3].

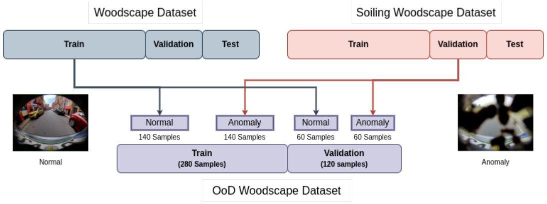

Figure 4: Dataset for training the OoD monitoring function

4. Early Experiments and Results

Dataset Building. For training the DNN model for semantic segmentation we used the Valeo Woodscape dataset [1] [27] with the semantic segmentation labels. For training the monitoring function (i.e., Bayesian generative classifier), our first choice was to use *Soiling Woodscape* sub-dataset. However, after inspecting the dataset, we noticed that samples were taken in small sequences. To improve dataset diversity and implement our approach, we decided to create a new smaller sub-dataset by taking just one or two samples from the sampling sequences for each anomaly in soiling Woodscape. We called this new dataset OoD Woodscape, and it combines samples from the Woodscape training set (normal class) and samples from the Soiling Woodscape validation set (anomaly class). The ooD-Woodscape training set has 280 samples, 140 samples for each class; the validation set has 120 samples total, 60 samples for each class. The dataset-building procedure is depicted in Figure 4.

Experiments. We quantify the entropy from intermediate latent vectors. Using the entropy values, we estimate the entropy density for each sub-dataset, i.e., samples from normal and anomaly sub-datasets. First, we quantify the entropy assuming a multivariate Gaussian distribution $\hat{H}_\phi(\mathbf{z} \mid x)$, as presented in Figure 5 top-right. Next, we compute the entropy estimation for each variable in the latent vector $\hat{H}_\phi(z_i \mid x)$, as shown in Figure 5-bottom. Finally, for comparison, we also use the *Mahalanobis* distance which is a multivariate measure of the distance between a point and distribution. In this last case, we built the reference distribution taking intermediate representations $\mathbf{z_i}$ for each input image x_i, from the Woodscape validation set (see Figure 5 top-left). Then, we measure the distance to this reference distribution using $d_M = \sqrt{(\mathbf{z}^* - \mu_{\mathbf{z_{val}}})^T \Sigma^{-1}_{\mathbf{z_{val}}} (\mathbf{z}^* - \mu_{\mathbf{z_{val}}})}$, for a new input image $\mathbf{x}^*$ and its predicted latent vector $\mathbf{z}^*$.

For entropy, in both cases, we observe that the densities for InD and OoD samples are different. In the first case, the estimated latent vector density shows clear multimodality for OoD samples, with peaks in entropy intervals that denote under-confident (uncertainty high) and overconfident (uncertainty very low) predictions. In the latter case, the entropy from latent vector variables, we observe that some variables exhibit multimodal density predictions for OoD samples and density peaks in different entropy value intervals from those obtained with InD samples. Finally, the d_M density shows slight peaks or modes for OoD samples, however, densities for InD and OoD have a high degree of overlap.

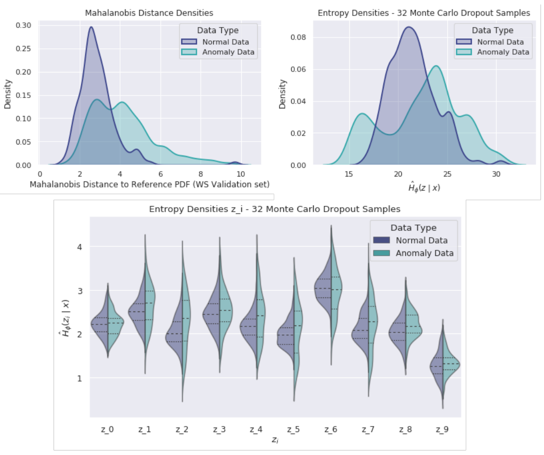

Figure 5: Illustration of empirical densities with KDE: Mahalanobis distance d_M (top-left), the multivariate Gaussian entropy $\hat{H}_\phi(\mathbf{z} \mid x)$ (top-right), and the entropy from latent each vector variable $\hat{H}_\phi(z_i \mid x)$.

Metrics. To evaluate our monitoring function, we used the validation set from OoD-Woodscape (the dataset we designed and built). We report the results using the following metrics, as suggested by Ferreira et al. [28] and Blum et al. [6]. In this regard, we report the Matthews correlation coefficient (MCC), the F1-score, the area under the Receiver Operating Characteristic (AUROC), and the False-Positive Rate at 90% True Positive Rate (FPR90) values. Table 1 summarizes the results used for each statistic or feature employed in our classifier (monitoring function), and Figure 6, shows the ROC curve.

Results & Discussion. We present the results of our monitoring function (classifier) in Table 1 and in Figure 6. In the results, we can see that the latent vector entropy-based methods outperform the Mahalanobis distance-based d_M method in almost all the performance metrics. We believe that the reason behind the poor performance of the d_M method is the strong assumption on the embedding space being class conditional Gaussian we building the reference distributions to compute the distance. On the hand, we can see that latent vector variable entropy has the best results. The reason behind the performance is that the classifier benefits from getting more expressive (entropy) information at the latent variable level.

[1] https://woodscape.valeo.com/download

Method	MCC	F1	AUROC	FPR90
d_M	0.473	0.763	0.769	0.5
$\hat{H}_\phi(\mathbf{z} \mid x)$	0.572	0.797	0.855	0.4
$\hat{H}_\phi(z_i \mid x)$	**0.685**	**0.849**	**0.946**	**0.16**

Table 1
Evaluation of OoD detection methods using DNN latent representations

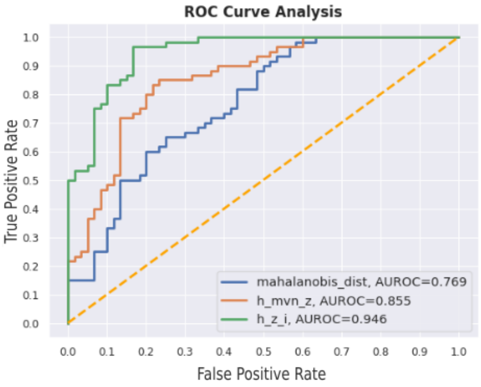

Figure 6: OoD detector ROC Curve analysis

5. Conclusion

In this work, we presented a method to use the uncertainty from intermediate latent representations for Out-of-distribution detection in a semantic segmentation task. Our early results show that using the entropy from latent features can be useful in building data-driven monitoring functions. In future work, we aim to explore the impact of the structure in the latent space by relaxing the Gaussian assumption [29] and its effect on the metrics and statistics used for the OoD detection task. Moreover, it is important to analyze the applicability of our approach in other semantic segmentation architectures that do not present generative blocks of neural networks.

Acknowledgement

This work has been supported by the French government under the “France 2030” program as part of the SystemX Technological Research Institute within the Confiance.ai Program (www.confiance.ai).

References

[1] R. McAllister, G. Kahn, J. Clune, S. Levine, Robustness to out-of-distribution inputs via task-aware generative uncertainty, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 2083–2089.

[2] F. Arnez, H. Espinoza, A. Radermacher, F. Terrier, A comparison of uncertainty estimation approaches in deep learning components for autonomous vehicle applications, Proceedings of the Workshop on Artificial Intelligence Safety 2020 (2020).

[3] J. Postels, H. Blum, Y. Strümpler, C. Cadena, R. Siegwart, L. Van Gool, F. Tombari, The hidden uncertainty in a neural networks activations, arXiv preprint arXiv:2012.03082 (2020).

[4] G. Di Biase, H. Blum, R. Siegwart, C. Cadena, Pixel-wise anomaly detection in complex driving scenes, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 16918–16927.

[5] Y. Xia, Y. Zhang, F. Liu, W. Shen, A. L. Yuille, Synthesize then compare: Detecting failures and anomalies for semantic segmentation, in: European Conference on Computer Vision, Springer, 2020, pp. 145–161.

[6] H. Blum, P.-E. Sarlin, J. Nieto, R. Siegwart, C. Cadena, The fishyscapes benchmark: measuring blind spots in semantic segmentation, International Journal of Computer Vision 129 (2021) 3119–3135.

[7] W. Morningstar, C. Ham, A. Gallagher, B. Lakshminarayanan, A. Alemi, J. Dillon, Density of states estimation for out of distribution detection, in: International Conference on Artificial Intelligence and Statistics, PMLR, 2021, pp. 3232–3240.

[8] Y. Sun, Y. Ming, X. Zhu, Y. Li, Out-of-distribution detection with deep nearest neighbors, arXiv preprint arXiv:2204.06507 (2022).

[9] K. Lee, K. Lee, H. Lee, J. Shin, A simple unified framework for detecting out-of-distribution samples and adversarial attacks, Advances in neural information processing systems 31 (2018).

[10] J. Nitsch, M. Itkina, R. Senanayake, J. Nieto, M. Schmidt, R. Siegwart, M. J. Kochenderfer, C. Cadena, Out-of-distribution detection for automotive perception, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, 2021, pp. 2938–2943.

[11] C.-L. Li, K. Sohn, J. Yoon, T. Pfister, Cutpaste: Self-supervised learning for anomaly detection and localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 9664–9674.

[12] S. A. Kohl, B. Romera-Paredes, C. Meyer, J. De Fauw, J. R. Ledsam, K. H. Maier-Hein, S. Eslami, D. J. Rezende, O. Ronneberger, A probabilistic u-net for segmentation of ambiguous images, arXiv preprint arXiv:1806.05034 (2018).

[13] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmen-

tation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.

[14] K. Sohn, H. Lee, X. Yan, Learning structured output representation using deep conditional generative models, Advances in neural information processing systems 28 (2015) 3483–3491.

[15] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, S. Udluft, Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning, in: International Conference on Machine Learning, PMLR, 2018, pp. 1184–1193.

[16] M. Henaff, Y. LeCun, A. Canziani, Model-predictive policy learning with uncertainty regularization for driving in dense traffic, in: 7th International Conference on Learning Representations, ICLR 2019, 2019.

[17] F. Arnez, H. Espinoza, A. Radermacher, F. Terrier, Improving robustness of deep neural networks for aerial navigation by incorporating input uncertainty, in: Computer Safety, Reliability, and Security. SAFECOMP 2021 Workshops, Springer International Publishing, Cham, 2021, pp. 219–225.

[18] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision?, in: Advances in neural information processing systems, 2017, pp. 5574–5584.

[19] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, J. Snoek, Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift, Advances in Neural Information Processing Systems 32 (2019) 13991–14002.

[20] E. Daxberger, J. M. Hernández-Lobato, Bayesian variational autoencoders for unsupervised out-of-distribution detection, arXiv preprint arXiv:1912.05651 (2019).

[21] A. Jesson, S. Mindermann, U. Shalit, Y. Gal, Identifying causal-effect inference failure with uncertainty-aware models, Advances in Neural Information Processing Systems 33 (2020) 11637–11649.

[22] Y. Gal, Z. Ghahramani, Dropout as a bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, 2016, pp. 1050–1059.

[23] G. Ghiasi, T.-Y. Lin, Q. V. Le, Dropblock: A regularization method for convolutional networks, Advances in Neural Information Processing Systems 31 (2018) 10727–10737.

[24] K. Deepshikha, S. H. Yelleni, P. Srijith, C. K. Mohan, Monte carlo dropblock for modelling uncertainty in object detection, arXiv preprint arXiv:2108.03614 (2021).

[25] L. Kozachenko, N. N. Leonenko, Sample estimate of the entropy of a random vector, Problemy Peredachi Informatsii 23 (1987) 9–16.

[26] J. VanderPlas, Python data science handbook: Essential tools for working with data, " O'Reilly Media, Inc.", 2016.

[27] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Chennupati, S. Nayak, S. Mansoor, X. Perrotton, P. Perez, Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[28] R. S. Ferreira, J. Arlat, J. Guiochet, H. Waeselynck, Benchmarking safety monitors for image classifiers with machine learning, in: 2021 IEEE 26th Pacific Rim International Symposium on Dependable Computing (PRDC), IEEE, 2021, pp. 7–16.

[29] P. Ghosh, M. S. Sajjadi, A. Vergari, M. Black, B. Scholkopf, From variational to deterministic autoencoders, in: International Conference on Learning Representations, 2019.