Evaluation of Uncertainty Estimation in Deep learning using Constraint-Based Dataset Generation

Deebul Nair
SpaceR Research Group
Luxembourg

Sathwik Panchangam
Institute for Artificial Intelligence and Autonomous Systems
Germany

Miguel A. Olivares-Mendez
SpaceR Research Group
Luxembourg

Nico Hochgeschwender
Department of Mathematics and Computer Science
Germany

Abstract

Uncertainty estimation in deep learning methods is a challenging problem because there is no true label available for uncertainty. Evaluation of the uncertainty estimation methods is a critical component for determining their practical utility, particularly when making decisions based on the predictions generated by a model. This paper identifies the shortcomings of existing techniques for assessing uncertainty estimation methods especially for embodied agent applications. To overcome these limitations, we introduce constraint-based dataset generation. Our methodology allows us to systematically evaluate the performance of different uncertainty estimation methods in a controlled and reproducible simulated environment. We generate subset of the datasets for image classification task based on constraints and then compare the predicted uncertainties between these subsets based on expert knowledge. We evaluated three different uncertainty estimation methods and reported on their differences. The proposed methodology should help in better understanding the uncertainty estimation capability of deep learning models deployed in embodied agents.

CCS Concepts

• **Computer systems organization** → **Reliability**; • **Computing methodologies** → **Uncertainty quantification**; Neural networks.

Keywords

Uncertainty estimation, synthetic dataset generation, DNN

SAC'25, March 31 –April 4, 2025, Sicily, Italy

ACM ISBN 979-8-4007-0629-5/25/03
https://doi.org/xx.xxx/xxx_x

ACM Reference Format:
Deebul Nair, Sathwik Panchangam, Miguel A. Olivares-Mendez, and Nico Hochgeschwender. 2025. Evaluation of Uncertainty Estimation in Deep learning using Constraint-Based Dataset Generation. In *Proceedings of ACM SAC Conference (SAC'25).* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/xx.xxx/xxx_x

1 Introduction

The accurate estimation of uncertainties in deep learning is a critical component for deploying them in real-world applications. This has led to significant interest in developing techniques to estimate uncertainty in deep neural networks [19] [1]. For embodied agents like autonomous car and robots, uncertainty estimation of deep learning models is particularly important as it can increase the dependability attributes like safety, reliability, robustness of the embodied agents, enabling better decision-making in complex environments where uncertainty is always present. However, evaluating the effectiveness of uncertainty estimation in deep learning is challenging due to the absence of ground truth data against which its performance can be measured.

Evaluation of the uncertainty estimation method in the absence of ground truth is a challenging problem. Uncertainty estimation methods are evaluated using uncertainty metrics or proper scoring rules [15] [8], benchmarking with Bayesian networks outputs [27], reliability evaluation with out-of-distribution data [24] [14] [13] and robustness evaluation with adversarial attack[24] [25] [16] or corrupted data [10] [9]. Even though these evaluation methods are good metrics for assessing the effectiveness of uncertainty estimation they are limited in evaluating only the performance, reliability and robustness. These evaluation methods fail to evaluate the effectiveness of the uncertainties in the case of their usage in embodied agents like autonomous car or domestic robots.

Deep neural networks (DNN) have become the de-facto method for perception in robotics due to their exceptional performance with high dimensional data. The output of these networks are utilized to make autonomous decisions for planning and control of the robots. Since the networks are stochastic in nature and also there is noise in sensors we use the probabilistic output of the networks to estimate the state of the environment. These probabilistic outputs are then used as the measurement model in filtering techniques such as the

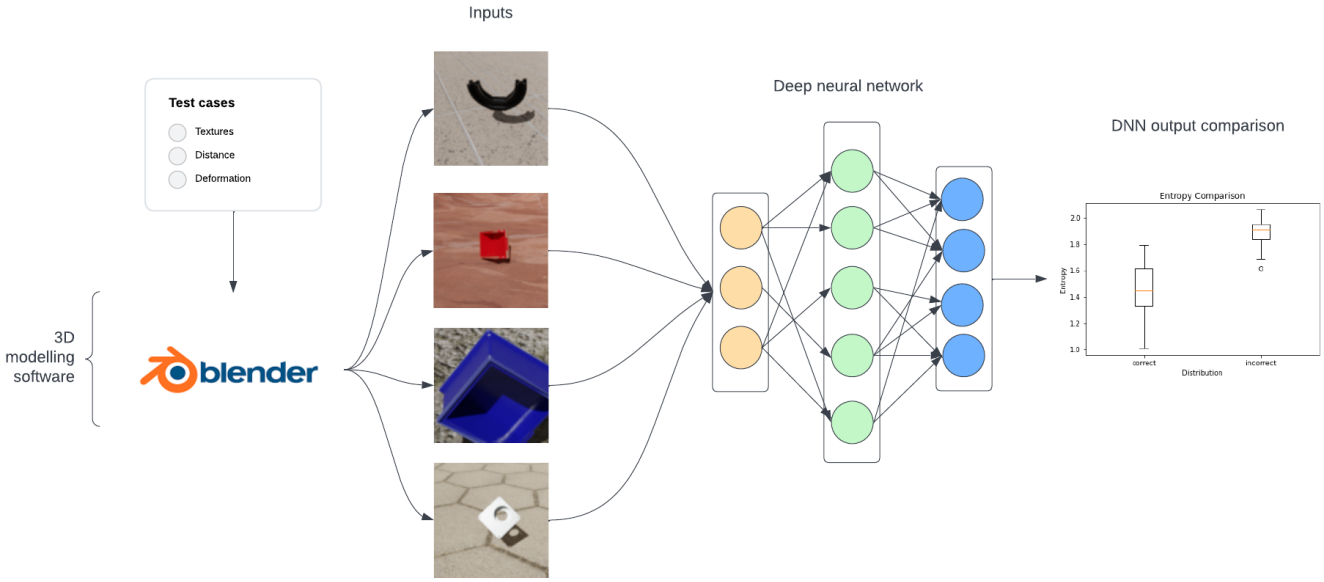

Figure 1: Concept overview of the evaluation of uncertainty estimated from DNN's in embodied agents for decision making and control.

Kalman filter [12] [22], Bayes filter [21] or particle filter [11], which can then be used for state estimation, decision-making under uncertainty and control applications. However, there are no proper evaluation methods to evaluate how the probabilistic outputs of the DNN can be utilized in statistical models, which poses a challenge for real-world applications of DNNs in robotics.

To address these limitations, our paper proposes a simulation-based dataset generation methodology that allows for the systematic evaluation of uncertainty estimation methods in a controlled and reproducible environment. We use the heuristics of an expert to determine a set of constraints that govern the distribution of uncertainties in a simulated dataset. Based on these parameters we generate a subset of the dataset which satisfies the specified constraint. The entropy's of the estimated uncertainty of these datasets generated under constraint are compared with the entropy's of the nominal dataset. This comparison gives a better understanding on how the uncertainty behaves in different scenarios the embodies agent perceives in the real world. Examples of constraints are distance of object from the camera, lightning condition of the environment, texture of the environment, deformability of the objects etc.

The contributions of the work are : *a*) We identify the limitations of current evaluation methods for deep learning models used in embodied agents (machines that interact with their environment). *b*) We propose a new method to generate datasets for evaluating these models using Blender, a 3D graphics software. *c*) We use these generated datasets to test and compare the performance of different uncertainty estimation methods for these models.

2 Limitation of evaluation methods of uncertainty estimation of DNN

In this section we discuss the different methods used to evaluate uncertainty estimation methods and their limitations with respect to their usage in embodied agents.

2.1 Uncertainty metric based evaluation

The most common method for evaluating uncertainties is by using the different uncertainty metrics. The most commonly used metrics for discrete outputs are Brier score [7] [15] [16], expected calibration error [4] [18], negative log-likelihood and [17] [26] [24]. Even though these metrics provide a quantitative measure of the estimated uncertainty, they fail to capture the entire range of possible outcomes and can be biased by model performance. The Brier score has the drawback of being influenced by the number of categories in the dataset [3] . In multi-class classification tasks, the Brier score may be biased towards models that predict a larger number of classes, even if their predictions are less accurate [23]. This can make it difficult to compare models with different numbers of classes. In case of expected calibration error the result depends upon the distribution of data across the number of bins used for calculation and the number of bins chosen affects the ECE algorithm [20]. This means that models with low ECE can still have poorly calibrated predictions or be overconfident in its predictions. Negative log likelihood gives more preference to the correctness of the output than to the uncertainty correctness [2]. We still can use these metrics to evaluate the performance, in this work we use entropy as a measurement of the uncertainty and use it to compare dataset subsets.

2.2 Robustness evaluation with adversarial and corrupted data

Another avenue for evaluating uncertainty estimation is by assessing the robustness of the DNN to adversarial attacks [24] [25] [16] and corrupted datasets [10] [9]. The expected behavior is that the uncertainty estimate of the adversarial attack dataset and the corrupted dataset is higher than the original, clean dataset. This is an example of using the heuristics of experts to evaluate uncertainty estimates. We expand on this idea by generating multiple subsets of clean and corrupted datasets which the embodied agent can see over its lifetime and use them as testing datasets.

2.3 Reliability evaluation with Out-of-Distribution (OOD) dataset

In addition to robustness evaluation, reliability evaluation with out-of-distribution datasets is another important aspect of the evaluation of uncertainty estimation [24] [14] [13]. In this evaluation, the expected outcome is that the uncertainty estimate of the OOD dataset is higher than that of the in-distribution dataset. This is also a valid assumption and is also an example of using expert heuristics to evaluate uncertainty estimates. We argue that for the reliability test not only out-of-distribution we also have to look into different subsets of in-distribution and verify the change of uncertainty.

3 Constraint Dataset Generation

We take inspiration from software testing and verification, where the human expert provides expected input and output to generate test cases that certify the performance of the software under various possible inputs. We expand on this idea to test uncertainty estimates of DNN's by generating constraint-based datasets that cover different scenarios that the DNN is expected to encounter during its operation. However, since the expected true uncertainty is not a quantifiable value it difficult for humans to provide the true expected uncertainty. However, humans are good with provided comparative judgments between uncertain situations, allowing us to generate pairs of datasets with varying degrees of uncertainty. For example, the uncertainty of objects which are far away from the camera is expected to be higher than objects nearer to it. We use simulation based dataset generator as it allows us to control various factors such as lighting conditions, object occlusion, and camera angle while generating datasets with varying degrees of uncertainty.

In real world, the embodied agent has to perceive the environment to make a decision. The environment has many variations like lighting, occlusion and scenarios where the images are blurry or the perceiving environment itself changes. For a deep learning model it is very much required to represent the uncertainty reliably in such scenarios to avoid failures and catastrophes. To represent the uncertainty in such scenarios the deep learning models must be evaluated on the datasets representing such scenarios. Collecting datasets for different real world scenarios is however not a suitable option because of the limitiations of time and cost. But all these scenarios

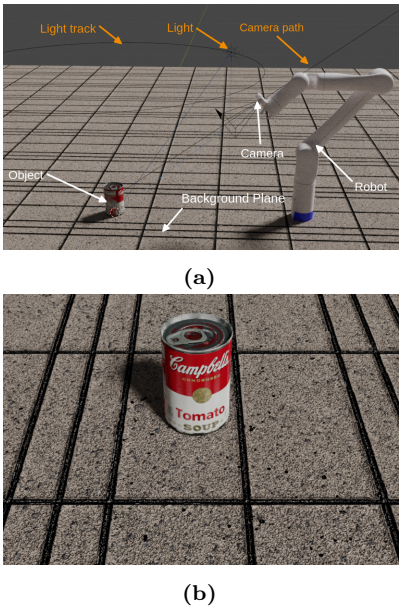

Figure 2: Blender based dataset generation: (a) Scene setup with robot and the different scene parameters. (b) Photo-realistic rendered image of the object. (c) Corresponding label with different embodiment information including distance between camera and object

can be simulated in a graphics software like blender and corresponding synthetic datasets can be generated.

In this methodology, we try to leverage the Blender software to create simulation based synthetic datasets. Blender [5] is a free and open-source 3D creation software used for creating 3D models, animations, and visual effects. Its features such as modeling, texturing, and rendering make it a powerful tool for creating realistic 3D environments and objects. It is used to develop dataset for training deep learning models [6]. Blender's ability to automate and randomize certain aspects of the dataset generation process makes it particularly useful for training deep learning models that require large and diverse datasets. We begin our approach by enumerating the heuristics for a specific constraint from a deep learning expert. The defined heuristics are used to identify a set of modifiable parameters in blender which are used to generate a nominal dataset and a subset of dataset which satisfies the specific constraint. In order to generate the datasets we need an environment, for this we create a scene in blender which consists of different components like, a background plane, a camera and a light source. By changing the parameters of these components such as intensity of light, focal length of camera, texture of background plane etc, we introduce variations to the datasets based on the heuristics defined by the expert. We create a nominal dataset with all variations of the environment and different subset of datasets which satisfy particular constraints. In the nominal dataset all the

Figure 3: Dataset generated with Blender under specific constraints a) Normal b) Far c) Dark d) Texture e) Deformed

objects present in the scene appear in normal conditions and there is all the variations introduced to the environment. On the other hand, for each subset of datasets we fix a particular variation and only generate those dataset which satisfies the constraint. For our experiments we generated 4 variations based on distance of object, lighting condition, background texture and deformation of objects. We selected a set of texture less industrial objects with 16 classes. All the objects had CAD models available which was the criteria for their selection.

3.1 Domain Specific Language

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

3.2 Blender meta-model

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor

gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

4 Experiments

In this work, we have selected four uncertainty estimation methods: cross-entropy model, MC dropout model [7], evidential deep learning model [24] and ensembles model [15]. We used ResNet18 architecture for the cross-entropy, dropout and evidential models and for the ensembles model we have averaged the results of four deep learning models. For training the four models we have used PyTorch framework keeping the hyper-parameters same for all the models. For all the experiments we used the step learning rate scheduler with a learning rate of 0.001. We have chosen the batch size as 128 and a weight decay of 1e5. In the following experiments all the four models are trained with normal dataset 3 and their performance is tested with the constraint datasets. For comparing and evaluating the expected behavior of the uncertainty estimation methods we compare the box plot of the entropy of the dataset. In the following experiments we expect the uncertainty of test constraints to be greater than the uncertainty of training constraint. Also, for a good uncertainty estimation method it is expected that the entropy of incorrect predictions to be high. This implies that there is no high confidence values being assigned for the miss classifications. Thus from the following experiments, in the boxplots we expect to observe a clear separation between the entropy of correct and incorrect predictions.

4.1 RQ1: Do predicted uncertainties change based on object distance?

To evaluate the performance of uncertainty estimation methods in the scenario of distance of object to camera, we have created two constraints far and near. The hypothesis for the impact of object's distance condition is that when the models trained on normal conditions and tested with the far distance dataset we expect the uncertainty to be high. To check this hypothesis, we have trained four uncertainty estimation methods on normal conditions dataset and test their performance on far constraint dataset. The entropy of both the dataset is plotted in Fig. 4 , in the case of cross-entropy, dropout and ensembles models, we can see that there is no separation for the entropy of correct and incorrect predictions of far distance constraint. Also, for the incorrect predictions these three models have low entropy values implying that the models are providing high confidence to the miss-classifications and thus their outputs cannot be trusted. We observe that the the entropy for correctly classified (orange bar) predictions for all uncertainty estimation methods is higher than the near dataset. This confirms that the models learn make changes to the uncertainty based on the distance. We also observe the entropy range of correctly classified far dataset is very much near to entropy of miss-classifed (red bar) for cross-entropy, dropout and ensembles method, while there is highest separation for the evidential loss function. This is of particular importance because based on the entropy of the miss-classified classes different decision making algorithms are written.

4.2 RQ2: Do predicted uncertainty change when lighting change?

To evaluate the performance of the uncertainty estimation methods on environmental lighting conditions, we have created the dark constraint dataset. The hypothesis for the impact of lighting conditions is that when the models trained on normal conditions and tested with dark lighting dataset, we expect the uncertainty to be high in the dark dataset. The entropy of both the dataset are plotted in Fig. 5.

We observe that our hypothesis is accepted for all the 4 methods. The entropy of the dark dataset is higher than the normal dataset. We also observe there is entropy of the correct predictions in the dark dataset is separated from the incorrect of the normal dataset.

4.3 RQ3: Do predicted uncertainty change when the background change?

To evaluate the performance of uncertainty estimation methods in the scenario of change in background textures, we have created a constraint dataset called textures. The image backgrounds present in textures dataset does not belong to the image backgrounds present in normal conditions dataset, thereby ensuring that the images present in textures constraint are not seen during the training process. The hypothesis for the impact of environmental background conditions

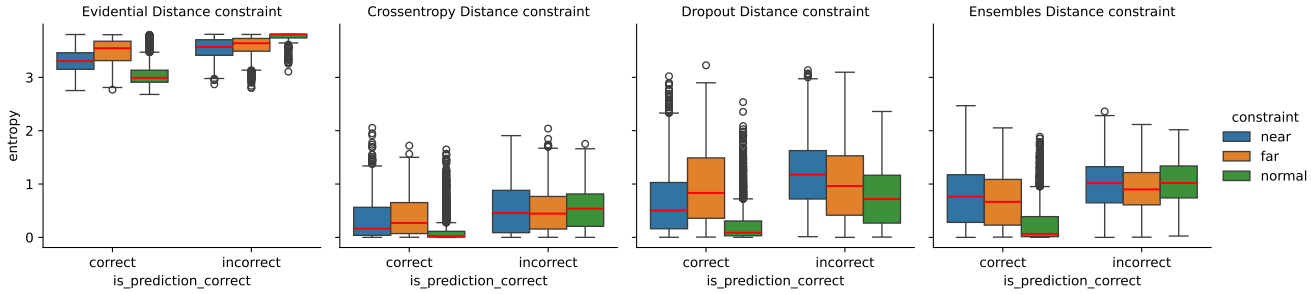

Figure 4: Entropy comparison of uncertainty estimation methods for nominal and far distance dataset.

is, when the models trained on normal conditions are tested with the images present in textures constraint we expect the uncertainty to be same as the model has never seen such textures before, as we are learning the object and not the background. To check this hypothesis, we have trained four uncertainty estimation methods on normal conditions dataset and test their performance on textures constraint dataset. From the figureFig. 6, Based on the observations we can accept the hypothesis for all the constraints. We observe there is minimal change of entropy for correct predictions when the background changes. We also observe ensemble has the highest change in entropy indicating ensembles also learn about the background.

5 Discussions

The advantage of the proposed methodology is that one can generate any new subset of the dataset based on a any new constraint. As uncertainty cannot be measured one can only evaluate the methods based on heuristics of human experts. The methodology enables such experts to develop test dataset and complete the evaluation. DNN's developed for perception of environments in embodied agents requires additional evaluation methods as compared to fixed environment tasks like medical diagnosis datasets. One limitation of the methodology is that one has to generate the scenes in Blender which are similar to real world scenes. We have open sourced the dataset generation code here https://github.com/DependableSoftware2-0/ConstraintBasedBlenderDatasetGenerator. Future work we would like to add additional constraints to the dataset generator.

6 Conclusions

Uncertainty estimates from the deep learning networks are used as observation models in different statistical models like filtering, state estimation and decision under uncertainty. The current evaluation methods focus majorly on robustness and reliability attributes but fail to evaluate their performance as required by statistical models. In this work, we focussed on addressing this gap and proposed an artificially generated dataset based on constraints to test the uncertainty estimates of deep neural networks. The proposed method generates this subset of a dataset based on a particular scenario that the DNN will observe when its deployed. For each subset, a human expert provided the expected uncertainty estimate behavior. The predicted and expected uncertainties are utilized to evaluate the performance of the DNN. We compared 3 state-of-art uncertainty estimation methods for the task of object classification of non-texture industrial objects. We trained the models using the dataset generated and for evaluating the uncertainties we generated 3 constraint-based dataset based on distance, lightning condition, background and deformation. For each constraint dataset, we mentioned the expected uncertainty and then used the entropy of the predictions to compare with the hypothesis. Based on our hypothesis we learned that the uncertainty estimation methods learn about lightning conditions and object distance to camera however the methods dont learn about the shape of objects. We hope the proposed methodology helps in a better understanding of uncertainty estimation methods.

7 Acknowledgments

Deebul Nair gratefully acknowledges the ongoing support of the Bonn-Aachen International Center for Information Technology and a PhD scholarship from the Graduate Institute of the Bonn-Rhein-Sieg University. This work was supported by the European Union's Horizon 2020 project SESAME (grant agreement No 101017258).

Acknowledgments

The authors would like to thank Dr. Yuhua Li for providing the matlab code of the *BEPS* method.

The authors would also like to thank the anonymous referees for their valuable comments and helpful suggestions. The work is supported by the National Natural Science Foundation of China under Grant No.: 61273304 and Young Scientsts' Support Program (http://www.nnsf.cn/youngscientsts).

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion* 76 (2021), 243–297.

[2] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry P. Vetrov. 2020. Pitfalls of In-Domain Uncertainty Estimation and Ensembling in Deep Learning. *ArXiv* abs/2002.06470 (2020).

[3] Melissa Assel, Daniel D Sjoberg, and Andrew J Vickers. 2017. The Brier score does not evaluate the clinical utility of diagnostic tests or prediction models. *Diagnostic and prognostic research* 1, 1 (2017), 1–7.

[4] Gustavo Carneiro, Leonardo Zorron Cheng Tao Pu, Rajvinder Singh, and Alastair Burt. 2020. Deep learning uncertainty and confidence calibration for the five-class polyp classification from colonoscopy. *Medical image analysis* 62 (2020), 101653.

[5] Blender Online Community. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam. http://www.blender.org

[6] Maximilian Denninger, Dominik Winkelbauer, Martin Sundermeyer, Wout Boerdijk, Markus Knauer, Klaus H. Strobl, Matthias Humt, and Rudolph Triebel. 2023. BlenderProc2: A Procedural Pipeline for Photorealistic Rendering. *Journal of Open Source Software* 8, 82 (2023), 4901. https://doi.org/10.21105/joss.04901

[7] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*. PMLR, 1050–1059.

[8] Tilmann Gneiting and Adrian E Raftery. 2007. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association* 102, 477 (2007), 359–378.

[9] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2020. AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)* (2020).

[10] Philipp Joppich, Sebastian Dorn, Oliver De Candido, Jakob Knollmüller, and Wolfgang Utschick. 2022. Classification and Uncertainty Quantification of Corrupted Data Using Supervised Autoencoders. In *Physical Sciences Forum*, Vol. 5. MDPI, 12.

[11] Peter Karkus, David Hsu, and Wee Sun Lee. 2018. Particle filter networks: End-toend probabilistic localization from visual observations. *arXiv preprint arXiv:1805.08975* (2018).

[12] Itzik Klein, Guy Revach, Nir Shlezinger, Jonas E. Mehr, Ruud J. G. van Sloun, and Yonina. C. Eldar. 2022. Uncertainty in Data-Driven Kalman Filtering for Partially Known State-Space Models. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 3194–3198. https://doi.org/10.1109/ICASSP43922.2022.9746732

[13] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. 2020. Being bayesian, even just a bit, fixes overconfidence in relu networks. In *International conference on machine learning*. PMLR, 5436–5446.

[14] Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. 2021. Learnable uncertainty under Laplace approximations. In *Uncertainty in Artificial Intelligence*. PMLR, 344–353.

[15] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30 (2017).

[16] Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. 2020. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems* 33 (2020), 7498–7512.

[17] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. 2020. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters* 5, 2 (2020), 3153–3160.

[18] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems* 32 (2019).

[19] José Mena, Oriol Pujol, and Jordi Vitria. 2021. A survey on uncertainty estimation in deep learning classification systems from a Bayesian perspective. *ACM Computing Surveys (CSUR)* 54, 9 (2021), 1–35.

[20] Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring Calibration in Deep Learning.. In *CVPR workshops*, Vol. 2.

[21] Johannes Pankert, Maria Vittoria Minniti, Lorenz Wellhausen, and Marco Hutter. 2021. Deep Measurement Updates for Bayes Filters. *IEEE Robotics and Automation Letters* 7, 1 (2021), 414–421.

[22] Guy Revach, Nir Shlezinger, Xiaoyong Ni, Adria Lopez Escoriza, Ruud JG Van Sloun, and Yonina C Eldar. 2022. KalmanNet: Neural network aided Kalman filtering for partially known dynamics. *IEEE Transactions on Signal Processing* 70 (2022), 1532–1547.

[23] David Rindt, Robert Hu, David Steinsaltz, and Dino Sejdinovic. 2022. Survival regression with proper scoring rules and monotonic neural networks. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1190–1205.

[24] Murat Sensoy, Lance Kaplan, and Melih Kandemir. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems* 31 (2018).

[25] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. 2020. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*. PMLR, 9690–9700.

[26] Bin Wang, Jie Lu, Zheng Yan, Huaishao Luo, Tianrui Li, Yu Zheng, and Guangquan Zhang. 2019. Deep uncertainty quantification: A machine learning approach for weather forecasting. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2087–2095.

[27] Andrew Gordon Wilson, Pavel Izmailov, Matthew D Hoffman, Yarin Gal, Yingzhen Li, Melanie F Pradier, Sharad Vikram, Andrew Foong, Sanae Lotfi, and Sebastian Farquhar. 2022. Evaluating approximate inference in Bayesian deep learning. In *NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 113–124.

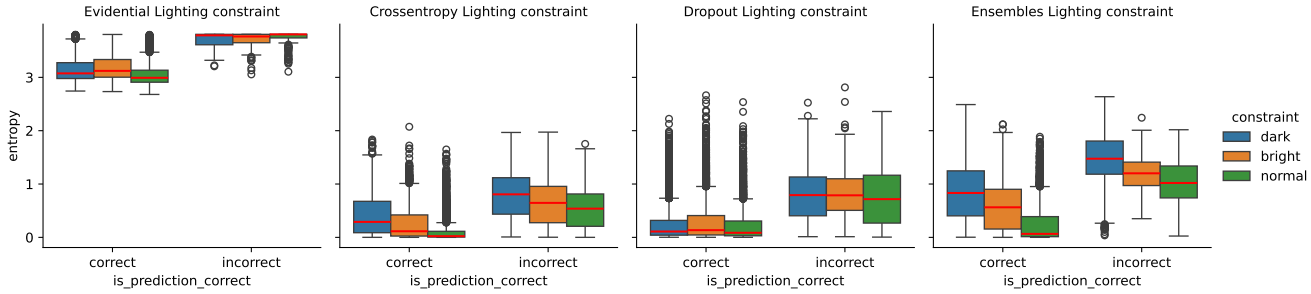

Figure 5: Entropy comparison of uncertainty estimation methods for nominal and dark lighting dataset.

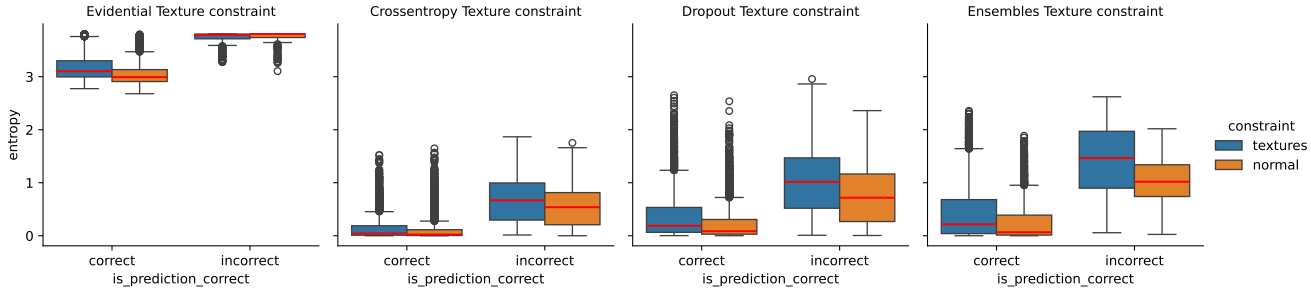

Figure 6: Entropy comparison of uncertainty estimation methods for nominal and texture dataset.