

# Bayesian Uncertainty Quantification with Synthetic Data

Buu Phan, Samin Khan, Rick Salay, and Krzysztof Czarnecki

University of Waterloo, Waterloo, Canada  
`{btphan,sa24khan}@uwaterloo.ca`  
`{rsalay,kczarne}@gsd.uwaterloo.ca`

**Abstract.** Image semantic segmentation systems based on deep learning are prone to making erroneous predictions for images affected by uncertainty influence factors such as occlusions or inclement weather. Bayesian deep learning applies the Bayesian framework to deep models and allows estimating so-called epistemic and aleatoric uncertainties as part of the prediction. Such estimates can indicate the likelihood of prediction errors due to the influence factors. However, because of lack of data, the effectiveness of Bayesian uncertainty estimation when segmenting images with varying levels of influence factors has not yet been systematically studied. In this paper, we propose using a synthetic dataset to address this gap. We conduct two sets of experiments to investigate the influence of distance, occlusion, clouds, rain, and puddles on the estimated uncertainty in the segmentation of road scenes. The experiments confirm the expected correlation between the influence factors, the estimated uncertainty, and accuracy. Contrary to expectation, we also find that the estimated aleatoric uncertainty from Bayesian deep models can be reduced with more training data. We hope that these findings will help improve methods for assuring machine-learning-based systems.

**Keywords:** Semantic Segmentation · Uncertainty · Influence factors

## 1 Introduction

Deep neural network (DNN) models, although having achieved many state-of-the-art results on a variety of tasks in computer vision [10, 17, 1], are not perfect as their performance much depends on the input images. Since errors in prediction due to the input characteristics are inevitable, several uncertainty metrics have been proposed as failure indicators for potentially faulty predictions [8, 6, 9]. Before deploying these metrics as failure indicators into safety-critical applications such as autonomous driving (AD) and medical diagnosis, their behaviour should be studied extensively under different scenarios in order to improve safety assurance.

Bayesian neural networks (BNN) is a class of deep learning models that is able to provide a more reliable uncertainty estimates than traditional DNN models. Also, these models can quantify two different types of uncertainty in supervised

learning problems, namely aleatoric and epistemic uncertainty. Aleatoric uncertainty represents the irreducible source of errors in the data (e.g., noise), and thus cannot be reduced by providing more data. In contrast, epistemic uncertainty represents the model’s “lack of knowledge” about the problem and can be reduced with more training data (more details in Sec. 3.2). Inputs with high aleatoric uncertainty can be thought as inherently ambiguous, whereas inputs with high epistemic uncertainty can be viewed as “unexpected”, i.e., far from the training dataset.

With recent concerns about AI safety in AD, researchers have applied BNN models to several vision tasks in AD, such as image segmentation [11, 12] and object detection [15], and reported that the obtained uncertainty estimates are more reliable. However, the experiments in these works do not investigate the effects of uncertainty influence factors [4], which are factors influencing the perceptual uncertainty, on the uncertainty estimates. A possible reason could be the lack of datasets with real-world images with varying influence factors and the associated factor labels. Yet understanding how the uncertainty estimates behave under these factors is needed for assuring the system performance.

In this paper, we propose using synthetic data to investigate and study the effects of selected factors on BNN’s uncertainty in the task of image semantic segmentation. In particular, we consider scene-specific factors: depth, occlusion, clouds, rain, and puddles. These factors can lead to ambiguous inputs, e.g., due to reduced information about the underlying objects in the scene because of their far distance or high occlusion level, but also unexpected inputs because of changed appearance, e.g., due to rain or puddles, if not trained for. We perform two sets of experiments, with the following results:

1. We study and quantify the uncertainty estimates of a state-of-the-art BNN under different levels of occlusion and depth for vehicles. As expected, we find a correlation between the influence factors, the estimated uncertainty, and accuracy, which is desirable for a failure indicator. Contrary to expectation, we find that the estimated aleatoric uncertainty from a BNN *can* change with more training data.
2. We explore and report on the behavior of BNN’s uncertainty estimates under different weather effects. We find that cloud level has much more significant impact on the uncertainty estimates than rain and puddles—which correlates with the higher negative impact of clouds on the network performance than that of rain or puddles, as previously reported for the same dataset and network [13].

The paper is structured as follows. Sec. 2 briefly describes the ProcSy dataset, which we use in this research. Sec. 3 explains the concept of BNN, aleatoric uncertainty, and epistemic uncertainty, and how to extract these uncertainties from BNN models. Sec. 4 describes the semantic segmentation network used in the experiments. Sec. 5 presents two experiments using ProcSy to study the effects of influence factor variations on the uncertainty estimates. Finally, we conclude the paper and suggest future research directions in Sec. 6.

## 2 The ProcSy Dataset

This section briefly describes the ProcSy synthetic dataset, which we have developed in previous work [13], and use for our experiments. Being synthetically generated, ProcSy holds several benefits for studying uncertainty estimation. Section 2.1 explains these benefits in detail. Section 2.2 summarizes the content of the ProcSy dataset.

### 2.1 Why Synthetic Data?

In the context of autonomous driving, factors such as depth, amount of occlusion, and weather effects can produce ambiguous and unexpected inputs to the model. Studying effects of these factors with a real-world dataset is difficult, although this is desirable. Current segmentation datasets with weather effects such as Raincouver [18] have a limited quantity of data to work with. Raincouver, for instance, is meant to be used as an addendum to an existing dataset such as Cityscapes rather than by itself, as it only contains 326 finely annotated images.

Berkeley Deep Drive [19] is a more recent dataset that shows more promise in the quantity of data availability (5683 finely annotated images). However, this dataset suffers from labeling inconsistency issues. These factors make real-world segmentation datasets impractical for model uncertainty analysis. Using these datasets to supplement a high-quality segmentation dataset such as Cityscapes [3] (which has no weather effects) is also problematic, since there is a qualitative difference in the datasets.

It is also very often the case that data acquired in the real world is not repeatable under different conditions. For instance, a scene captured in ideal conditions may not be reproducible during a day with heavy rain, because a car that was originally parked in the scene is no longer there. This sort of logistical issues can prove to be very expensive to overcome in generating a real-world dataset. It is also not an easy task to annotate effects such as amount of occlusion in a real-world dataset. On the other hand, synthetic data rendered by recent computer graphics technology can provide various influence factor effects with minimal labour cost. Due to these reasons, in this paper, we use our ProcSy synthetic dataset to analyze the effect of influence factors on the uncertainty estimates of a model.

### 2.2 Dataset Summary

Our ProcSy dataset is comprised of road scenes captured from a virtual render of a  $3 \text{ km}^2$  map region of urban Canada. From this environment, 11,000 scene frames were curated to contain no visible artifacts such as clipping of the camera through vehicles and pedestrians. These curated frames were then split into 8000 training frames, 2000 validation frames, and 1000 test frames.

For each frame of the ProcSy dataset, along with the base RGB image, we have generated corresponding ground truth annotation labels, depth map data, and occlusion maps of the different vehicle types present in the scene.

We generate weather variations in the categories of rain, puddle, and cloud coverage. For each of these factors, we consider five different intensity levels. These are 0%, 25%, 50%, 75%, and 100%. Fig. 1 shows an example frame with different intensity levels for each of the three weather factors. For our training set of 8000 images, we first consider three equal subsets for the weather categories. Within each subset, we further divide into intensity levels and render RGB images reflecting variations in these subsets. This allows us to carry out experiments without having to generate every permutation of influence factor variations.



**Fig. 1.** Weather variations visualizing intensity level differences in the three weather categories — rain, cloud, and puddle; ground truth, depth map, and occlusion map for one vehicle type are also shown along with the base RGB image

### 3 Uncertainty Estimation with Bayesian Neural Networks

In this section, we describe the concept of a BNN, two types of uncertainty (aleatoric and epistemic), and how to extract measures of these types of uncertainty from a BNN’s predictions. We focus on the case of classification since we are interested in uncertainty estimates of the image segmentation task.

#### 3.1 Bayesian Neural Networks with MC-dropout

With BNN models, instead of getting a single prediction from a learned set of weight values, we obtain the prediction by taking into account the outputs of multiple models, whose weight values are derived under the Bayesian framework. In regions with a large amount of data, the predictions of these models are consistent with each other; on the other hand, in regions that lack data, this

consistency does not tend to hold. Since deep models contain a large number of weights, applying the Bayesian framework to a deep model is computationally intractable, therefore, in order to obtain different sets of weight values, we need to use Bayesian approximation techniques.

One approach to obtain an approximated BNN model from an existing DNN architecture is by inserting dropout layers and training the new model with dropout training [8]. At test time, for a given input, we perform multiple forward predictions in the network while keeping the dropout layers active. In other words, we remove a percentage of randomly-selected units (i.e., set the weight values of their connections to 0) from the trained model in order to obtain a sample prediction for the given input; then we repeat this process  $T$  times and calculate the average prediction. This technique at test time is referred to as Monte-Carlo (MC) dropout.

Specifically in classification, we are given the input data  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  and the associated labels  $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ , where each  $\mathbf{y}_j$  belong to one of the  $K$  classes  $[1, 2, \dots, K]$ . We use  $p(\hat{\mathbf{y}} = k|\hat{\mathbf{x}}, \omega_i)$  to denote the probability that the label  $\hat{\mathbf{y}}$  of the test input  $\hat{\mathbf{x}}$  is the  $k$ th class, which is given by a model (trained with dropout on  $\mathbf{X}, \mathbf{Y}$ ) with a set of sampled weight values  $\omega_i$  as its softmax output. Then, we wish to capture the mean probability  $p(\hat{\mathbf{y}} = k|\hat{\mathbf{x}})$  for a test point  $\hat{\mathbf{x}}$ , which can be calculated as:

$$p(\hat{\mathbf{y}} = k|\hat{\mathbf{x}}) = \frac{1}{T} \sum_{i=1}^T p(\hat{\mathbf{y}} = k|\hat{\mathbf{x}}, \omega_i) \quad (1)$$

where  $T$  is the number of MC-dropout samples and  $\omega_i$  is a set of weight values for each MC-dropout sample. We note that calculating the probability for every class will give us a categorical distribution over classes for the input  $\hat{\mathbf{x}}$ .

### 3.2 Types of Uncertainty

Aleatoric uncertainty represents the irreducible noise in the data and cannot be reduced even when we gather more data [12]. For example, in the binary classification problem where we have a large amount of data, the data that lie within the intersection region of the two class distributions will have higher aleatoric uncertainty than the data in either distribution but outside the intersection. Epistemic uncertainty captures the model's lack of knowledge due to the limitation in the training data (such as bias, scarcity, novelty, etc.) [7]. It can be reduced by gathering more training data. Bayesian modelling allows us to quantify both types of uncertainty. Although approaches exist for estimating aleatoric uncertainty with non-Bayesian approaches [12], they cannot be used to estimate epistemic uncertainty. Furthermore, non-Bayesian models tend to perform poorly and give overconfident predictions in regions that lack data [7]. Bayesian approaches to neural networks, on the other hand, allow us to capture the epistemic uncertainty [12] and, thus, BNN models tend to give predictions with high uncertainty in low-density regions.

### 3.3 Uncertainty Estimation

There are three metrics that we will use in the experiments, namely: predictive entropy  $\mathbb{H}(\hat{\mathbf{y}}|\hat{\mathbf{x}})$  (captures total uncertainty), mutual information  $\text{MI}(\hat{\mathbf{y}}|\hat{\mathbf{x}})$  (captures epistemic uncertainty), and aleatoric entropy  $\text{AE}(\hat{\mathbf{y}}|\hat{\mathbf{x}})$  (captures aleatoric uncertainty). These uncertainty estimates can be calculated as follows [5, 16].

**Predictive Entropy** captures the total amount of uncertainty (epistemic and aleatoric) and is equal to the sum of mutual information and aleatoric entropy. It is calculated as an entropy of the mean categorical distribution:

$$\begin{aligned}\mathbb{H}(\hat{\mathbf{y}}|\hat{\mathbf{x}}) &= \text{MI}(\hat{\mathbf{y}}|\hat{\mathbf{x}}) + \text{AE}(\hat{\mathbf{y}}|\hat{\mathbf{x}}) \\ &= -\sum_k p(\hat{\mathbf{y}} = k|\hat{\mathbf{x}}) \log p(\hat{\mathbf{y}} = k|\hat{\mathbf{x}})\end{aligned}\quad (2)$$

**Aleatoric Entropy** captures the aleatoric uncertainty and is calculated by averaging the entropy of each sampled categorical distribution.

$$\text{AE}(\hat{\mathbf{y}}|\hat{\mathbf{x}}) = -\frac{1}{T} \sum_{k,i} p(\hat{\mathbf{y}} = k|\hat{\mathbf{x}}, \omega_i) \log p(\hat{\mathbf{y}} = k|\hat{\mathbf{x}}, \omega_i) \quad (3)$$

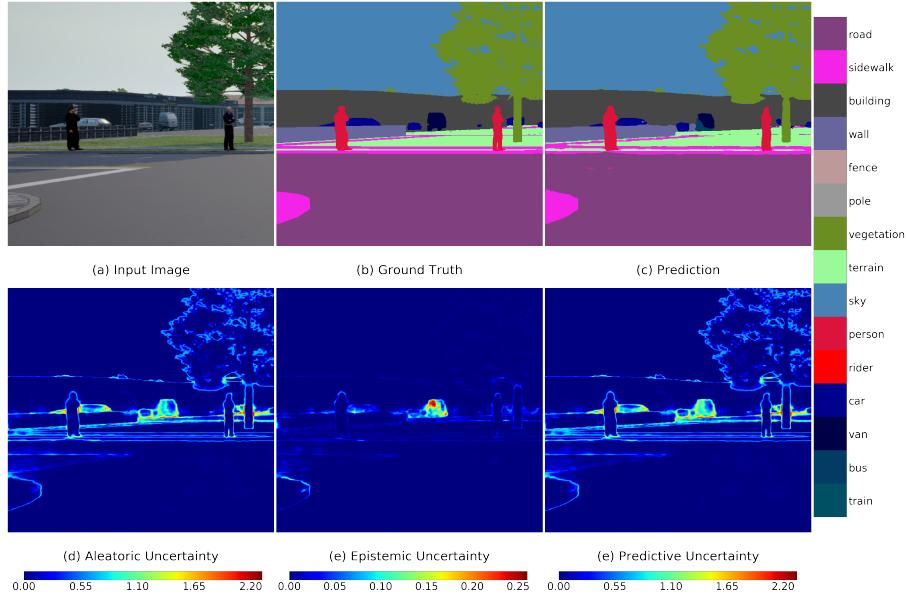
**Mutual Information** captures the epistemic uncertainty and is calculated as:

$$\text{MI}(\hat{\mathbf{y}}|\hat{\mathbf{x}}) = \mathbb{H}(\hat{\mathbf{y}}|\hat{\mathbf{x}}) - \text{AE}(\hat{\mathbf{y}}|\hat{\mathbf{x}}) \quad (4)$$

### 3.4 Bayesian Neural Networks for Image Segmentation

Semantic segmentation is a task that assigns a class label to each pixel of an image. For autonomous driving, the image segmentation system enables the vehicle to perceive the visual state of the world. Since deep convolutional architectures consider this task as classifying each pixel independently using the same network [14], the BNN approach can be applied to this family of architectures to estimate the uncertainty per pixel.

For the experiments in this paper, we use Deeplab v3+ [2], one of the state-of-the-art models for segmentation, with ResNet 50 backbone architecture [10]. We inserted dropout layers with rate of 0.5 at four blocks in the middle of the backbone (specifically, at the end of the 8<sup>th</sup> till 11<sup>th</sup> block). The basis for this setup is based on the studies by Kendall et al. [11] and Mukhoti et al. [16], from which they empirically determined that inserting dropout layers in the middle flow yields a better predictive performance than for other positions in a network. Fig.2 shows examples of the three uncertainty types estimated by a BNN.



**Fig. 2.** Illustration for different types of uncertainty estimates in semantic segmentation. (a),(b),(c) show the input image, ground truth and prediction, respectively. (d),(e),(f) show the estimated aleatoric, epistemic and predictive uncertainty from our model. It can be visually observed that the aleatoric uncertainty is high at the class boundary (e.g., the tree). On the other hand, the epistemic uncertainty estimates are high only at several specific regions, such as the cluster in the middle of (e). This model is trained with 3000 clean images (model A in Sec.4).

## 4 Experiments with Synthetic Data

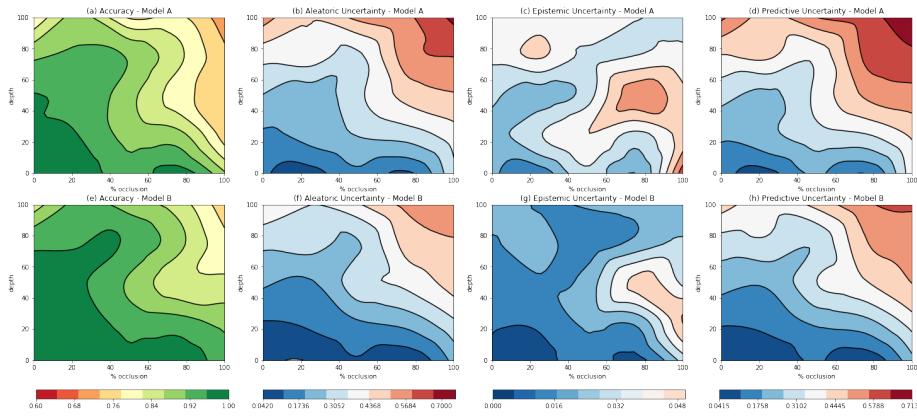
In this section, we perform two experiments with two sets of influence factors. The first experiment involves the amount of depth and occlusion as factors, whereas the second experiment involves different weather effects, specifically: clouds, rain, and puddles. The reason why we treat them separately is that occluded and distant objects occur in the training set, whereas the latter factors do not.

We train two Bayesian Deeplab v3+ models: model A with 3000 clean images and model B with 8000 clean images. We note that the set of 3000 clean images is a subset of the set of 8000 images. Model A and model B are trained with 75,000 and 180,000 iterations, respectively, with a batch size of 16 and crop size of 512x512.

### 4.1 Occlusion and Depth

In this experiment, we study how different variations of occlusions and depth factors affect the uncertainty estimates. We test and measure the uncertainty

estimates (aleatoric, epistemic, and predictive) of model A and B using a test set consisting of 270 clean images containing a total of 1,200 vehicles. The amount of occlusion for each vehicle is determined by calculating the fraction of the number of occluded pixels over the total number of the vehicle's pixels. We then assign this occlusion level to each visible pixel of the vehicle. For each model, we partition the pixels into subsets based on amount of occlusion and distance (each discretized into five intervals of 20%). Then we calculate the mean accuracy and uncertainty estimates for the model predictions of the pixels in each subset. Finally, we use cubic spline interpolation to obtain a contour plot. The results are shown in Fig.3.

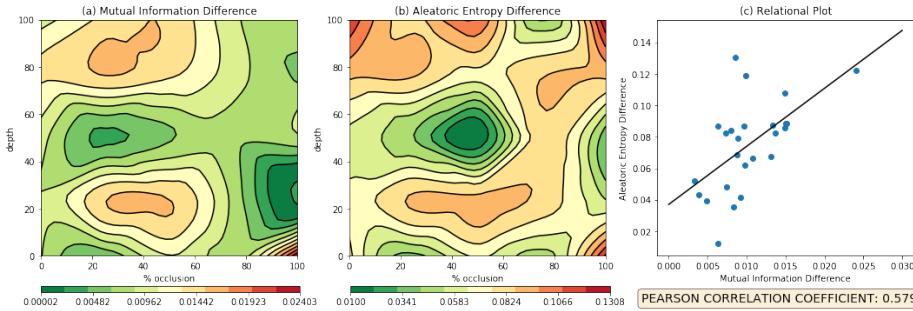


**Fig. 3.** The two rows show the accuracy, aleatoric, epistemic and predictive uncertainty estimates according to level of depth and occlusion of model A and B, respectively. Each color bar reflects the metric values of the plots in the corresponding column.

According to the definition of aleatoric and epistemic uncertainty, we expect that model B's epistemic uncertainty estimates should be lower than model A's, whereas the aleatoric estimates of the two models should stay similar. Based on the results in Fig. 3, we make the following observations:

1. Model B has higher accuracy and lower epistemic uncertainty than model A in general. This fits with expectations since model B is trained with more data than model A.
2. The predictive uncertainty can be observed to be correlated well with the accuracy, which is a desirable characteristic for a failure indicator. The Pearson correlation coefficient between predictive uncertainty and accuracy for model A is  $-0.89$  and for model B is  $-0.90$ .
3. For both models A and B, the aleatoric estimates increase for objects that are more occluded and further away from the camera as expected. The same behavior can also be observed for the predictive uncertainty estimates.
4. There is a difference between the aleatoric estimates of the two models, which is surprising. Specifically, the difference between aleatoric estimates seems to

increase with the epistemic difference. To validate this observation, we plot in Fig. 4 the difference between those two uncertainty estimates according to the amount of occlusion and depth, then we calculate the Pearson correlation coefficient of these quantities. The results reflect this observation as the Pearson coefficient, which values is 0.579, implies that there is a relation between the two quantities.



**Fig. 4.** (a) shows the difference between the estimated mutual information for model B and A according to occlusion and depth. (b) shows the equivalent difference for aleatoric entropy. (c) shows the relational plot between the two differences. Each blue dot represents the difference between the aleatoric and epistemic estimates in a certain subset of occlusion and depth. The black line, which shows the relation between the two variables, is fitted by using linear regression.

We hypothesize that the reason why we observe the aleatoric uncertainty estimate changing when more training data is provided is that the estimate is only reliable where we have sufficient data. For regions with a sufficient amount of data, the decision boundary of model A and B are similar to each other. Thus, adding more data would not likely change the estimated aleatoric uncertainty (unless the data we already have is biased). On the other hand, the decision boundaries in regions that lack data tend to be inconsistent making the aleatoric estimate unreliable.

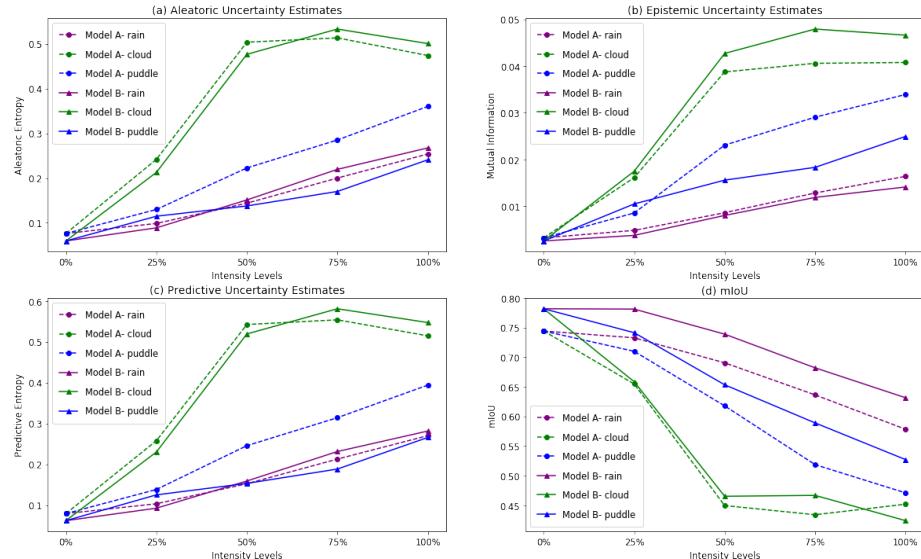
Finally, we note that for model B, high aleatoric estimates still occur for regions that have relatively low epistemic uncertainty, such as the middle top region in Fig. 3f,g where the objects are occluded around 50% and far away from the camera. This suggests that depth and occlusion are sources of aleatoric uncertainty in the image segmentation task.

To ensure perceptual performance in safety-critical application, developers must make sure that the training data satisfies the scenario coverage condition properly [4]. This experiment and analysis suggests that we can use the measure of epistemic uncertainty to determine the optimal amount of data to collect for occlusion and depth factors. Specifically, we should collect enough data to make the epistemic uncertainty map blue. We leave the further exploration and validation of this idea for future work.

## 4.2 Weather

In this experiment, we study how the uncertainty estimates vary with respect to different intensity levels of weather effects, namely: clouds, rain, and puddles. We expect that as we increase the effect's intensity, the BNN model should have worse performance and output higher uncertainty estimates. This is because high intensity effects will introduce more artifacts that would cause misclassification in the model, thus the uncertainty estimate should increase to indicate this.

For each model and effect, we calculate the mIoU (mean Intersection over Union) and the mean aleatoric, epistemic, and predictive uncertainty estimates per pixel at every intensity level. Each effect's intensity level contains 150 images. The reason why we compare the performance of model A and B is that we want to observe how the uncertainty estimates change when we train with more in-distribution data. The results are shown in Fig. 5.



**Fig. 5.** (a-d) show the estimated aleatoric, epistemic, and predictive uncertainties and mIoU values for different variations of weather, factors respectively. The x-axes represent different intensity levels.

In terms of mIoU (Fig. 5d), we see that model B, which is trained with more in-distribution data, has better performance than model A when there is no factor involved. Further, model B's mIoU is higher than model A's for different intensity levels of rain and puddles. However, surprisingly for cloudiness, there is a sharp degradation in terms of mIoU for model B. Critically, at the 100% cloud intensity level, model A outperforms model B (although with just a small margin). We find that in 100% cloud conditions, the two models fail to predict the following classes: pole, traffic lights, sky, bicycle, car and truck.

In terms of uncertainty, the three types of uncertainty estimates, in general, increase with the intensity levels for every factor. This behavior meets our expectation for the uncertainty and it applies to both model A and B. We notice that for clouds, there is a small decrease for the 50% to 100% levels, which requires further investigation.

We also make two following observations. First, for model B, we see that the epistemic uncertainty corresponds to the mIoU better than the other two uncertainties. Specifically, at every intensity level for each effect, the ascending order for epistemic estimates are rain, puddle and cloud, which corresponds to the descending of that order in mIoU. For model A, on the other hand, all the uncertainties reflect the mIoU well. Second, there is an inconsistency in terms of the difference between model A and B’s uncertainty estimates for all the factors. For example, at 100% intensity level for rain and puddle, model B has lower epistemic uncertainty estimates than model A, yet it is higher in the case of cloudy images. This is unexpected since we assumed that training with more in-distribution data can only make the epistemic uncertainty lower or intact.

## 5 Conclusions and Future Work

Reliable uncertainty estimation of ML predictions is important for the safe use of ML-based components. In this paper, we use the ProcSy dataset to study the effects of different influence factors, namely: depth, occlusion, rain, clouds and puddles, on the uncertainty estimates of the BNN model for image segmentation.

In the experiments with occlusion and depth factors, our results show that the aleatoric uncertainty estimates are dependent on the epistemic uncertainty estimates. When given enough data, the epistemic uncertainty estimates reduce but the aleatoric estimates remain high for distant and occluded objects. Furthermore, we find that cloud affects the uncertainty estimates and mIoU values of the BNN more profoundly than rain and puddle, even when we have more in-distribution training data.

As we have mentioned in Sec. 4a, the experiment results suggest one potential application that requires further investigation: the possibility to use the epistemic uncertainty estimates to find an optimal amount of data for the occlusion and depth factors. Furthermore, it would be beneficial to extend this work to understanding the risk potential of these factors. Finally, while this paper studies how the influence factors affect the Bayesian uncertainties, future work should address how we can use this synthetic framework to evaluate the reliability of any uncertainty estimation method.

## References

1. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
2. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 801–818 (2018)

3. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
4. Czarnecki, K., Salay, R.: Towards a framework to manage perceptual uncertainty for safe automated driving. In: SAFECOMP 2018 Workshops, WAISE, Proceedings. pp. 439–445. Springer (2018)
5. Depeweg, S., Hernandez-Lobato, J., Doshi-Velez, F., Udluft, S.: Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In: 35th International Conference on Machine Learning, ICML 2018. vol. 3, pp. 1920–1934 (2018)
6. DeVries, T., Taylor, G.W.: Learning confidence for out-of-distribution detection in neural networks. arXiv preprint arXiv:1802.04865 (2018)
7. Gal, Y.: Uncertainty in deep learning (2016)
8. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning. pp. 1050–1059 (2016)
9. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 1321–1330. JMLR. org (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. arXiv preprint arXiv:1511.02680 (2015)
12. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? In: Advances in neural information processing systems. pp. 5574–5584 (2017)
13. Khan, S., Phan, B., Salay, R., Czarnecki, K.: Procsy: Procedural synthetic dataset generation towards influence factor studies of semantic segmentation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2019), to appear.
14. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
15. Miller, D., Nicholson, L., Dayoub, F., Sünderhauf, N.: Dropout sampling for robust object detection in open-set conditions. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 1–7. IEEE (2018)
16. Mukhoti, J., Gal, Y.: Evaluating bayesian deep learning methods for semantic segmentation. arXiv preprint arXiv:1811.12709 (2018)
17. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
18. Tung, F., Chen, J., Meng, L., Little, J.J.: The raincouver scene parsing benchmark for self-driving in adverse weather and at night. IEEE Robotics and Automation Letters **2**(4), 2188–2193 (2017)
19. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: BDD100K: A diverse driving video database with scalable annotation tooling. CoRR **abs/1805.04687** (2018), <http://arxiv.org/abs/1805.04687>