COMPARISON OF UNCERTAINTY QUANTIFICATION METHODS FOR CNN-BASED REGRESSION

K. Wursthorn*, M. Hillemann, M. Ulrich

Karlsruhe Institute of Technology (KIT), Institute of Photogrammetry and Remote Sensing (IPF), Karlsruhe, Germany
(kira.wursthorn, markus.hillemann, markus.ulrich)@kit.edu

Commission II, WG II/7

KEY WORDS: Deep Learning, Regression, Convolutional Neural Network, Uncertainty Quantification, Bayesian Modelling, Variational Inference

ABSTRACT:

The evaluation of reliability is not only of high importance for safety-critical deep learning applications but for object pose estimation as well. The uncertainty of the result is one way to express its reliability. In order to better understand existing uncertainty quantification (UQ) methods and their performance on image-based regression tasks, we use a small CNN and various scenarios to evaluate the estimated uncertainties. The evaluation is done on different simplistic synthetic datasets, consisting of gray-scale images of squares on a darker background. We train the CNN to predict the square center position of the square in the image. We compare how different UQ methods perform under dataset shift, rotation, occlusion, noise changes in the images.

1. INTRODUCTION

The increasing exploitation of deep-learning-based methods in real world applications requires the evaluation of the reliability of the model prediction results. For applications such as autonomous driving (McAllister et al., 2017) or the analysis of medical imaging results (Leibig et al., 2017) where the predictions are relevant for the safety of road users or the patient, uncertainty estimates are essential. In a more industrial setting, the knowledge of the uncertainty, for example, can be used for anomaly detection or the manipulation of a robot arm and is an important feature of human-robot collaboration (Huber, 2020). Furthermore, uncertainty estimates can be used in active learning (Gal et al., 2017) and for the detection of adversarial attacks. To localize and grasp an object successfully, a robot relies on the object pose. This entails estimating the translational and rotational parameters of the object in the robot coordinate system. The object pose for vision-guided robots is estimated from images. Aside from identifying a certain object in an image, the estimation of object poses with deep learning entails the regression of the object pose either in image coordinates or in camera coordinates. Recent computer-vision-based 6D object pose estimation approaches, which achieve state-of-the-art results on benchmark datasets, use deep learning models such as convolutional neural networks (CNN) to obtain the object pose (Hodan et al., 2020). However, it was observed that these models do not perform well for changes in the input data and are therefore difficult to use in mission-critical applications (Shi et al., 2021, Amodei et al., 2016, Loquercio et al., 2020). This motivates the desire to use uncertainty quantification (UQ) methods for image based regression tasks with convolutional neural networks.

The so-called predictive uncertainty of the predictions of a deep learning model is divided into the aleatoric uncertainty caused by noise in the input data and the epistemic uncertainty caused by the model weight parameters themselves due to a lack of knowledge (Abdar et al., 2021). Hence, the aleatoric uncertainty is also known as data uncertainty while the epistemic uncertainty is sometimes called model uncertainty. The aleatoric uncertainty follows the usual definition of statistical uncertainty and describes the random effects that affect the predictions of a model (Hüllermeier and Waegeman, 2021). According to whether the noise is constant over the input data or varies from data point to data point, the aleatoric uncertainty is further divided into homoscedastic or heteroscedastic (Kendall and Gal, 2017). While the aleatoric uncertainty describes the influence of the stochastic input data noise, and therefore cannot be reduced with more training data, the epistemic uncertainty increases for new input data not seen in the training process (Kendall and Gal, 2017). Therefore, the epistemic uncertainty is a systematic uncertainty that results from a lack of knowledge (Hüllermeier and Waegeman, 2021). Thus, epistemic uncertainty plays an important role for safety-critical applications where it is crucial to identify unknown situations.

In this paper, we compare different UQ methods for CNN-based regression. Previous contributions have evaluated such methods on small neural networks or on existing deep models and large datasets. We aim to achieve a better understanding of existing uncertainty quantification approaches when applied to image-based deep learning regression tasks with CNNs by evaluating them independently of large image datasets and complex deep model architectures. To reduce complexity, and hence to assess the effects of various influences, we use a simplistic synthetic image dataset and a straightforward architecture with relatively few weights.

After a brief describtion of some existing UQ methods in Section 2, the evaluation strategy of these methods is described in Section 3. The results of the evaluation are discussed in Section 4, before we draw a conclusion in Section 5.

2. RELATED WORK

Existing UQ methods are based on Bayesian modelling where a deterministic model is transformed into a stochastic one in order to achieve probabilistic predictions. In a stochastic neural

* Corresponding author

network, probability distributions are placed over either the model parameters or the layer activations (Jospin et al., 2022). Stochastic model weights lead to epistemic uncertainty estimation (Loquercio et al., 2020). The deterministic weights ω are replaced by the prior probability distribution $p(\omega)$. Given a dataset $\mathbf{X}, \mathbf{Y}$, with the training inputs $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ and the corresponding output targets $\mathbf{Y} = \{\mathbf{y}_1, ..., \mathbf{y}_N\}$, the posterior weights distribution after the training can be modelled with Bayes' theorem (Gal, 2016):

$$p(\omega|\mathbf{X}, \mathbf{Y}) = \frac{p(\mathbf{Y}|\mathbf{X}, \omega)\, p(\omega)}{p(\mathbf{Y}|\mathbf{X})}. \quad (1)$$

The probabilistic prediction result $\mathbf{y}^*$ of a new input data point $\mathbf{x}^*$ can be obtained by Bayesian inference (Gal, 2016):

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathbf{X}, \mathbf{Y}) = \int p(\mathbf{y}^*|\mathbf{x}^*, \omega)\, p(\omega|\mathbf{X}, \mathbf{Y})\, d\omega. \quad (2)$$

The analytical solution of Equation (2) as well as the Bayesian inference of the model evidence

$$p(\mathbf{Y}|\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{X}, \omega)\, p(\omega)\, d\omega \quad (3)$$

in Equation (1) are intractable due to the high dimensionality of the integration over the weights space (Gal, 2016, Kendall and Gal, 2017, Loquercio et al., 2020). This leads to approximation approaches known as variational inference methods sampling from either the posterior weights distribution, like Bayes-by-Backprop (Blundell et al., 2015) or the distribution of the predictions, like Monte-Carlo Dropout (Gal and Ghahramani, 2016), to compute the epistemic uncertainty. Variational inference is built on the idea of approximating the unknown posterior weights distribution by a simpler distribution $q(\omega)$. Bayes-by-Backprop (Blundell et al., 2015) does variational inference by assuming Gaussian weight distributions and updating both the weights as well as their variances in the backpropagation during the model training. This leads to the doubled amount of memory needed for loading the obtained Bayesian network, a draw-back that motivates the approach of (Gal and Ghahramani, 2016) to exploit a common regularization technique in deep learning to sample directly from the predictive distribution in Equation (2).

Monte-Carlo Dropout (Gal and Ghahramani, 2016) can be used for epistemic uncertainty estimation. It uses dropout regularization (Srivastava et al., 2014) commonly used during training in deep learning. By inserting a dropout layer after each weight layer (e.g., convolutional layer) of the model and activating dropout at inference time, the model outputs varying prediction results for the same input data. Multiple forward passes of the same input data are used to get T samples of predictions. These samples can be used to get the final prediction result by computing the mean value of the T sampled predictions as well as the uncertainty by estimating the standard deviation of the sample. The estimated uncertainty depends on the size of the sampled set of predictions and the dropout rate. As the method requires multiple passes through the model, the computational cost of the inference time is multiplied. This has to be taken into account while choosing a sample size T for real-time applications. Other methods that estimate the predictive uncertainty use Monte-Carlo Dropout for epistemic uncertainty estimation and combine this with aleatoric uncertainty estimation methods like assumed density filtering (Gast and Roth, 2018, Loquercio et al., 2020) or training with a log-likelihood loss function (Kendall and Gal, 2017).

A similar approach to Monte-Carlo Dropout is Deep Ensembles (Lakshminarayanan et al., 2017), but it looks at uncertainty estimation from perspective of a frequentist. Deep Ensembles train a set of models with the same underlying network architecture and the same prediction target and data set. Each of these models is used simultaneously to obtain a prediction for the same input data. Estimating the mean and standard deviation of the obtained prediction yields the final prediction result and model uncertainty. In (Lakshminarayanan et al., 2017), the best results were obtained with a negative log-likelihood loss function and virtual adversarial training. With regard to the uncertainty estimation in deep learning 6D object pose estimation, (Shi et al., 2021) used a small ensemble of pose estimation models to get the uncertainty of the predicted object poses.

Both Monte-Carlo Dropout and Deep Ensembles have additional computational cost. Due to sampling multiple predictions, the former approach leads to a multiple of the inference time of a single forward pass. Deep Ensembles however, uses a multiple of the memory of a single model to load the whole ensemble simultaneously. To reduce the computational cost and still achieve the high quality uncertainty estimates of Deep Ensembles, (Durasov et al., 2021) proposes fixed dropout masks to simulate multiple models while requiring one forward pass only.

In (Gast and Roth, 2018), the aleatoric uncertainty is estimated by propagating the noise of the input data through the model with assumed density filtering (ADF). In contrast to the aleatoric uncertainty estimation in (Kendall and Gal, 2017) where only the last layer considers stochastic activations, ADF assumes all intermediate layer outputs as stochastic values.

3. EVALUATION STRATEGY

To evaluate the UQ methods described in Section 2, we generate various synthetic simplistic datasets of gray-scale images. Each image depicts a square at a specific position in the image. The models are trained to predict the square center positions in the images. For the evaluation of the epistemic uncertainty estimation, we consider different evaluation scenarios that are summarized in Figure 1. Three of the evaluation scenarios are designed to evaluate three cases of new, unseen data that lead to a shift of the evaluation data compared to the training data:

1. images with squares that are far from the central region and thus outside the position distribution of the training data
2. images with different gray values than the training data, which are outside the gray value distribution of the training data
3. images with squares that are rotated or occluded and thus outside the training data distribution.

In (Candela et al., 2009), the dataset shifting of only the distribution of the targets $\mathbf{Y}$ in case 1 is referred to as prior probability shift, while the shifted distribution of the inputs $\mathbf{X}$ in

case 2 is called covariate shift. These cases describe scenarios that may have an influence on the epistemic uncertainty estimation. The aleatoric uncertainty estimation is done by changing the noise in the input images. With regard to industrial application settings and 6D object pose estimation in particular, where usually the same camera for image aquisition is used, we only consider homoscedastic noise, meaning constant input noise throughout all images, in our evaluation scenario.

The datasets shown in Figure 1 that are used to evaluate the the three scenarios and the aleatoric uncertainty estimation are described in Section 3.1. The architecture of the base CNN is modified if necessary and trained accordingly (section 3.2) on the training datasets and evaluated on the corresponding evaluation dataset. Different evaluation metrics are used (Section 3.3) as well as the visualization of the results.

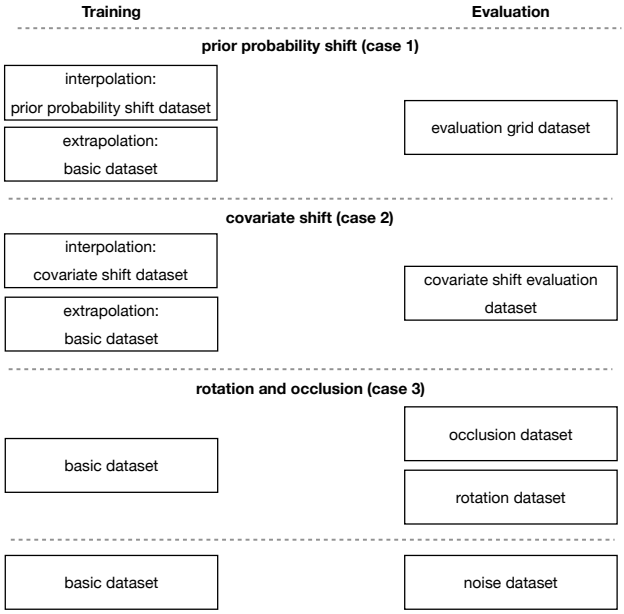


Figure 1. Evaluation strategy. We define three different evaluation scenarios for epistemic uncertainty estimation: Case 1 refers to a dataset shift only of the target square center positions, case 2 considers a dataset shift of the gray value distribution of the input data, and case 3 evaluates the epistemic uncertainty estimation in case of rotated and occluded squares in the evaluation images. The aleatoric uncertainty estimation is evaluated by increasing the input data noise in the evaluation images.

3.1 Datasets

As shown in Figure 1, we generate different datasets for training the models and the evaluation of UQ methods. All datasets consist of gray-scale images of 200×200 pixels, each depicting one square of constant gray value and size on a darker background at varying positions. Some examples of the basic dataset are shown in Figure 2(a). The overall goal of the trained models is the prediction of the square center position in image coordinates. To balance the target space, the origin of the image coordinate system is moved to the image center which corresponds to normalizing the square center positions. Additionally, we add zero-mean Gaussian noise with a standard deviation of two gray values to all images for homoscedastic noise and more realistic input images.

3.1.1 Basic Dataset All models are trained on the basic dataset. The square center positions of the images used for model training are sampled randomly from a given distribution. Figure 2(b) shows the training dataset with square center positions sampled from a Gaussian distribution. The red and blue dots represent the split between training and validation data. The basic dataset has 700 training images, 300 validation images, and 300 test images. By choosing a mean of $0\,\mathrm{px}$ and a standard deviation of $18\,\mathrm{px}^2$, the square center positions are mostly restricted to the central region of the image, as shown in Figure 2(b).

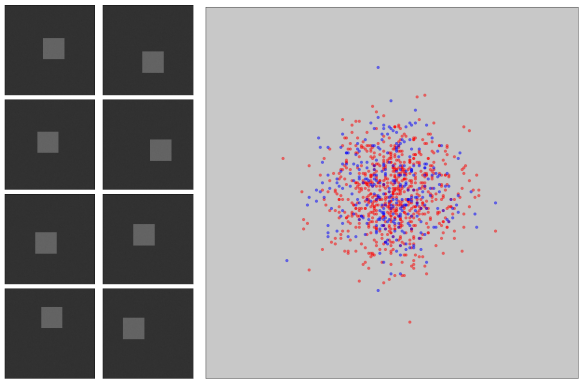
3.1.2 Evaluation Grid Dataset To evaluate the UQ methods on the whole image, we generate a dataset where the target square center positions are evenly distributed over the image plane. This lead to an evaluation grid. Hence, we call this dataset the evaluation grid dataset. More specificly, we choose a grid where each pixel of the image plane has a corresponding square center position. This means, for every pixel there is an image depicting a square centered at that pixel. With an image size of 200×200 pixels, this leads to an evaluation grid dataset with 40000 evaluation examples. Note that the gray-scale distribution is the same as in the basic dataset.

3.1.3 Dataset Shift Training and Evaluation Datasets Both cases can be interpreted as a lack of knowledge and therefore cause an increasing epistemic uncertainty. Case 1 as well as the evaluation of the uncertainty estimation on the basic datasets covered by the evaluation grid dataset. In case 2, the squares of the evaluation images are placed at identical positions in the image center but with varying gray values. These images are generated by adding various gray values to the entire original image. Therefore, the gray value distributions of these images have little to no overlap with the original distribution. This evaluation dataset is called the covariate shift evaluation dataset. The training data distribution depicted in Figure 2(b) (red dots) is used to evaluate UQ methods with respect to extrapolation. The square center positions of the training dataset of the prior probability shift dataset of case 1 and interpolation are composed of a cluster at the central region of the image and a ring of positions. Figure 3 shows the distributions of the training and validation data. As shown in Figure 3(a), for testing interpolation with the covariate shift dataset of case 2, we train the models on both darker and lighter images compared to the basic training data images depicted in Figure 2(a) and evaluate them on the covariate shift evaluation dataset, like we do in case of extrapolation. This is possible due to the range of the gray value shifts covered in the evaluation dataset. The training dataset for interpolation of case 2 is called the covariate shift dataset. As the dataset has large differences in the input data, the dataset consists of 7000 training images, 3000 validation images and test images. In principle, deep learning models are known to be able to generalize or interpolate but not to be able to extrapolate to unseen data. Consequently, we expect the trained models to show better prediction performance in case of interpolation compared to extrapolation. Therefore, we expect higher epistemic uncertainty estimates outside the training data in both cases.

3.1.4 Rotation Dataset To evaluate how the models and UQ methods cope with rotated squares in the images, the rotation dataset is generated. It consists of 360 images of squares rotated with increasing integer angles in the interval of $[0, 360)$ degrees. It is used for the evaluation of models that were trained an the basic dataset.

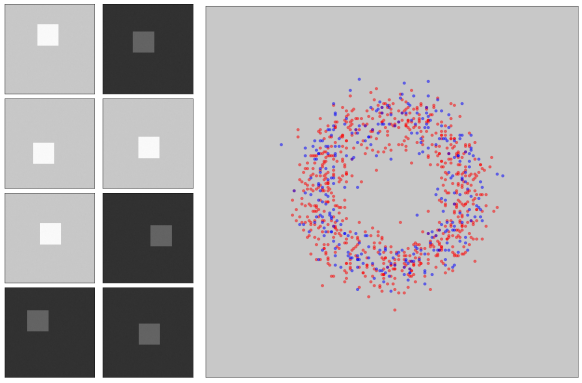
3.1.5 Occlusion Dataset Similar to the rotation dataset and the covariate shift evaluation dataset, examples to test the uncertainty estimation in case of partially occluded squares are generated. The images contain squares with various degrees of occlusion.

3.1.6 Noise Dataset In contrast to the other evaluation datasets, this dataset is used for evaluating the aleatoric uncertainty estimates. The dataset contains images of various gray-scale levels of Gaussian noise. The evaluated models are trained on the basic training dataset.



(a) (b)

Figure 2. (a) Examples from the training dataset. (b) Distributions of the square center positions in the training (red) and validation (blue) dataset.



(a) (b)

Figure 3. (a) Examples from the training dataset. (b) Distributions of the square center positions in the training (red) and validation (blue) dataset used for model training.

3.2 Models and Methods

The generated datasets are used to evaluate the UQ methods described in Section 2: Monte-Carlo Dropout (Gal and Ghahramani, 2016), Deep Ensembles (Lakshminarayanan et al., 2017) and the combined method of (Kendall and Gal, 2017) which we refer to as MCDONLL in the following. MCDONLL also uses Monte-Carlo Dropout for epistemic uncertainty estimation but trains a single model with the negative log-likelihood (NLL) loss function. Following the visualization of the different training and evaluation datasets in Figure 1, a CNN is trained for each method and each training dataset accordingly. The overall performance of the models is measured on an evaluation dataset of the same distributions as the training datasets.

The CNN with Monte-Carlo Dropout is referred to as MCDO, the Deep Ensemble as Ensemble, where five models are trained using the NLL loss function, and the approach of (Kendall and Gal, 2017) as MCDONLL. The model ADF uses the method proposed in (Gast and Roth, 2018) for aleatoric uncertainty estimation. Accordingly to (Loquercio et al., 2020), we use the trained weights of MCDO for MCDOADF.

3.3 Evaluation Metrics

The performance of each trained CNN is evaluated with RMSE metric. It should be mentioned that it is not the focus of this submission to find the best architecture for predicting square center positions. Instead, we aim to use simple architectures that can predict the square center position with a RMSE of less than 1.0 pixel.

Another suitable metric is the explained variance score (EVA) (Loquercio et al., 2018, Loquercio et al., 2020):

$$\mathrm{EVA} = 1 - \frac{\mathrm{Var}\left(\mathbf{y}_{true} - \mathbf{y}_{pred}\right)}{\mathrm{Var}\left(\mathbf{y}_{true}\right)} \tag{4}$$

where $\mathbf{y}_{true}$ are the targets and $\mathbf{y}_{pred}$ the predicted square center positions. Higher the EVA values indicate better results. The metric describes essentially how well a model captures the variance inherent in a dataset.

As it is usually impossible, to get ground truth uncertainty values, the NLL metric is often used to evaluate the quality of the uncertainty estimates (Gal and Ghahramani, 2016, Gast and Roth, 2018, Kendall and Gal, 2017, Loquercio et al., 2020):

$$\mathrm{NLL} = \frac{1}{2}\log\left(\sigma^2\right) + \frac{1}{2\sigma^2}\left(y_{true} - y_{pred}\right)^2 \tag{5}$$

Here, σ^2 is the estimated variance to be evaluated. The lower the NLL metric the better is the uncertainty estimation of a method.

Due to the nature of the synthetic datasets and the prediction goal of the square center positions, we are able to estimate a ground truth variance or ground truth standard deviation of the uncertainty estimates and use it as evaluation metric. The absolute standard deviation is defined as

$$\mathrm{STD} = \frac{1}{N}\sum_{i=1}^{N}\left(y_{true} - y_{pred}\right)^2. \tag{6}$$

In contrast to the standard deviation estimates of the UQ methods that are relative standard deviations, the STD is computed with respect to the ground truth square center positions.

3.4 Implementation Details

We use the PyTorch framework for the implementation of the CNNs to get predictions. The model consists of three convolutional layers and a fully connected output layer. The convolutional layers have 16, 32, and 64 channels and kernel sizes of 9×9, 5×5, and 3×3, respectively. To reduce the influence of the weights in the fully connected output layer in comparison to the filter kernels of the convolutional layers on the predictions,

the size of the intermediate feature maps is reduced gradually by global average pooling with kernel sizes of $(2, 2)$, $(4, 4)$, and $(5, 5)$ in the convolutional layers. We use ReLU activation functions and batch normalization after each convolutional layer. For MCDO, the CNN is modified by inserting dropout layers after each convolutional layer. For MCDONLL using the NLL metric as loss function, the CNN uses an additional fully connected layer, trained to predict the logarithmic variance. Otherwise, the mean squared error loss function is used. For ADF, we used the code provided by (Gast and Roth, 2018). The models were trained using the Adam optimizer (Kingma and Ba, 2015) and a constant learning rate. All models are trained to achieve roughly similar predictive quality in terms of RMSE on the evaluation dataset. We do not aim to achieve the best possible prediction results and focus on the comparability of the methods instead.

4. RESULTS AND DISCUSSION

The results on the basic dataset are summarized in Table 1. The table shows that all models predict the square center positions with a RMSE in subpixel range on the test data of the basic training dataset. The very high EVA values confirm that the models are able to adapt well to the variability in the simplistic datasets. In terms of NLL, it can be observed that this metric is higher for the epistemic uncertainty methods MCDO and Ensemble compared to the aleatoric uncertainty methods ADF and NLL.

Model	RMSE [px]	EVA [%]	NLL
ADF	0.046	100.0	−0.85
NLL	0.028	100.0	−1.55
MCDO	0.266	99.9	0.79
Ensemble	0.004	100.0	−2.18
MCDOADF	0.260	99.9	0.15
MCDONLL	0.495	99.8	0.14

Table 1. Results on the test data of the basic dataset. Higher values for RMSE and EVA and lower values for NLL indicate better results.

In the following, we present the results of the evaluation scenarios described in Section 3. First, the results of the cases 1, 2, and 3 and their influence on UQ methods that estimate epistemic uncertainty are shown in Section 4.1. In Section 4.2, the influence of increased input noise on the UQ methods for aleatoric uncertainty estimation are shown.

4.1 Epistemic Uncertainty

4.1.1 Prior Probability Shift Figure 4 shows the uncertainty results of the models MCDO, MCDONLL, and Ensemble on the evaluation grid dataset for the prior probability shift in case 1 and extrapolation. MCDOADF is not depicted because the results are essentially identical to the results of MCDO as for both Monte-Carlo Dropout is used for epistemic uncertainty estimation. It shows for each pixel position the epistemic standard deviation estimates in column and row direction in gray values. The red dotted polygon is the convex hull of the training data, depicted in Figure 2(b). All three models show standard deviations of the same magnitude, with the highest predicted epistemic standard deviation for Ensemble, the lowest for MCDONLL. All three cases show similar patterns of high and low standard deviation estimates. Low epistemic uncertainties are obtained for evaluation images in which the squares are positioned inside the training data distribution. Starting with a low epistemic uncertainty at the central region of the image and inside the training data distribution, the standard deviation estimates increase for square center positions at the edges of the training data and decrease again at the image borders and in the corners. Outside but still near the training data distribution, the computed standard deviation values reflect the increased uncertainty of the model the further the evaluation data is shifted away from the training data. However, in the evaluation images with squares at the image borders and in the corners, where the squares are truncated partially and therefore unknown to the model, the model predicts the square center positions confidently near the mean of the training distribution causing low uncertainty estimates. Consequently, in this case, the estimated epistemic uncertainty does not correspond to the actual reliability of the model.

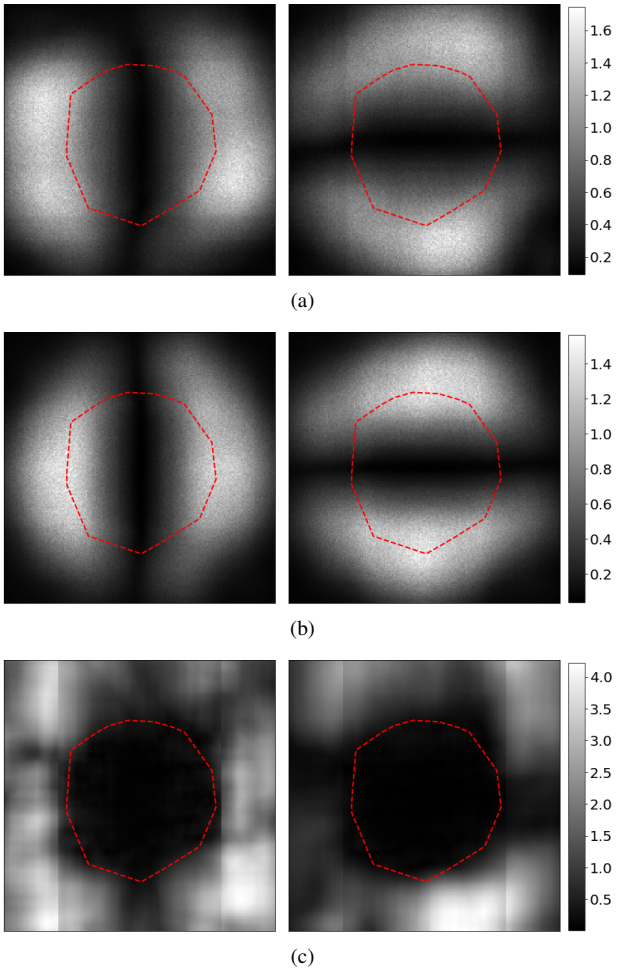


Figure 4. Epistemic standard deviation in column (left) and row (right) direction on the evaluation grid dataset in pixel units with (a) MCDO (Gal and Ghahramani, 2016) , (b) MCDONLL (Kendall and Gal, 2017) and (c) Ensemble (Lakshminarayanan et al., 2017) in case of prior probability shift (case 1) and extrapolation. The red dotted polygon represents the convex hull of the square center positions of the training dataset.

Figure 5 shows the results of (a) MCDO, (b) MCDONLL, and (c) Ensemble in case of interpolation. The models are trained on the prior probability shift dataset. The results in this case are similar to the ones shown in Figure 4 in case of extrapolation. The results of MCDO show increasing standard deviation estimates in the central region of the image where no training data

is available. MCDO and MCDONLL estimate standard deviations of the same magnitude. However, in contrast to MCDO, MCDONLL estimates lower standard deviations in the central region of the image. This could be explained by the different training loss functions that indirectly have an impact on the epistemic uncertainty estimation. The results of Ensemble (Figure (c)) show low standard deviation estimates inside the training data distribution compared to the estimates outside of the training data, including the central region of the image.

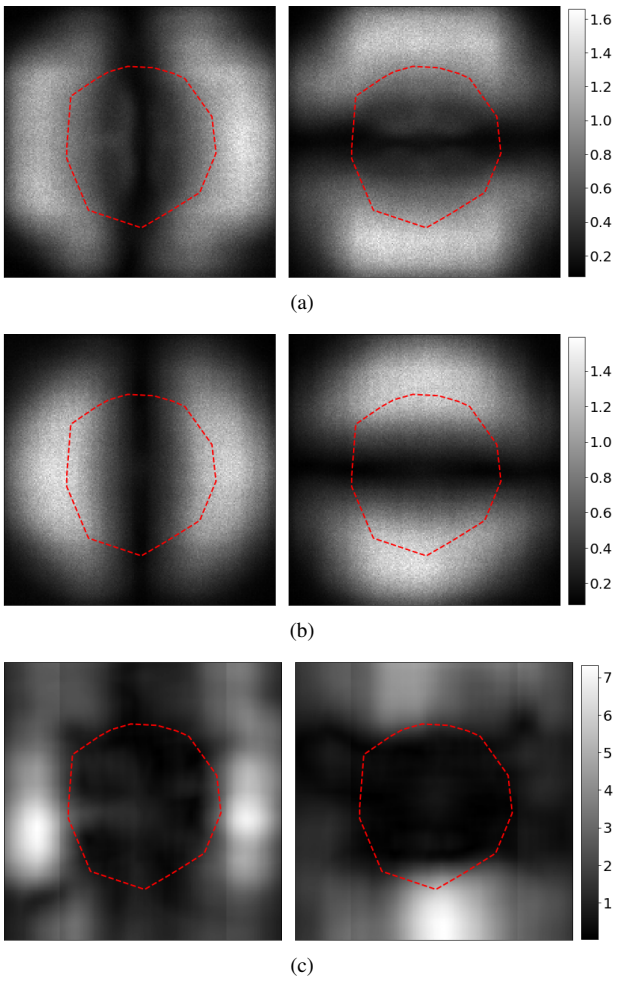


Figure 5. Epistemic standard deviation on column (left) and row (right) direction on the evaluation grid dataset in pixel units with (a) MCDO (Gal and Ghahramani, 2016) , (b) MCDONLL (Kendall and Gal, 2017), and (c) Ensemble (Lakshminarayanan et al., 2017) in case of case of prior probability shift (case 1) and interpolation. The models are trained on the prior probability shift dataset. The red dotted polygon represents the convex hull of the square center positions of the training dataset.

4.1.2 Covariate Shift The results of MCDO and Ensemble for the covariate shift (case 2) are depicted in Figure 6. The mean value of ten predictions is shown for each shifted dataset example. The epistemic uncertainty increases the more the gray value distribution is shifted from the original distribution. The results of MCDONLL, shown in Figure 6(b), are similar to the results of MCDO but differ more for column and row and are generally higher. The results shown in Figure 6(a) are in agreement with the experiments in (Gal and Ghahramani, 2016): With increasing distance of the input data from the training data distribution the uncertainty estimates increase as desired. However, there is an offset between the reference values of the absolute standard deviation and the estimates of the epistemic standard deviation. This can be explained by the uncalibrated nature of the estimates with Monte-Carlo Dropout (Gal and Ghahramani, 2016). In contrast, the epistemic standard deviation estimates of Ensemble in Figure 6(c) are in good agreement with the corresponding absolute standard deviation value. It can be noted that the standard deviation of Ensemble does not increase linearily with the gray value shift. Instead, Ensemble seems to be able to predict the square center positions even for large covariate shifts and only shows increasing standard deviations starting around a gray value shift of 100 levels. The predicted epistemic standard deviation seems to increase slower than the absolute standard deviation which leads to an underestimation of the epistemic uncertainty for large covariate shifts.

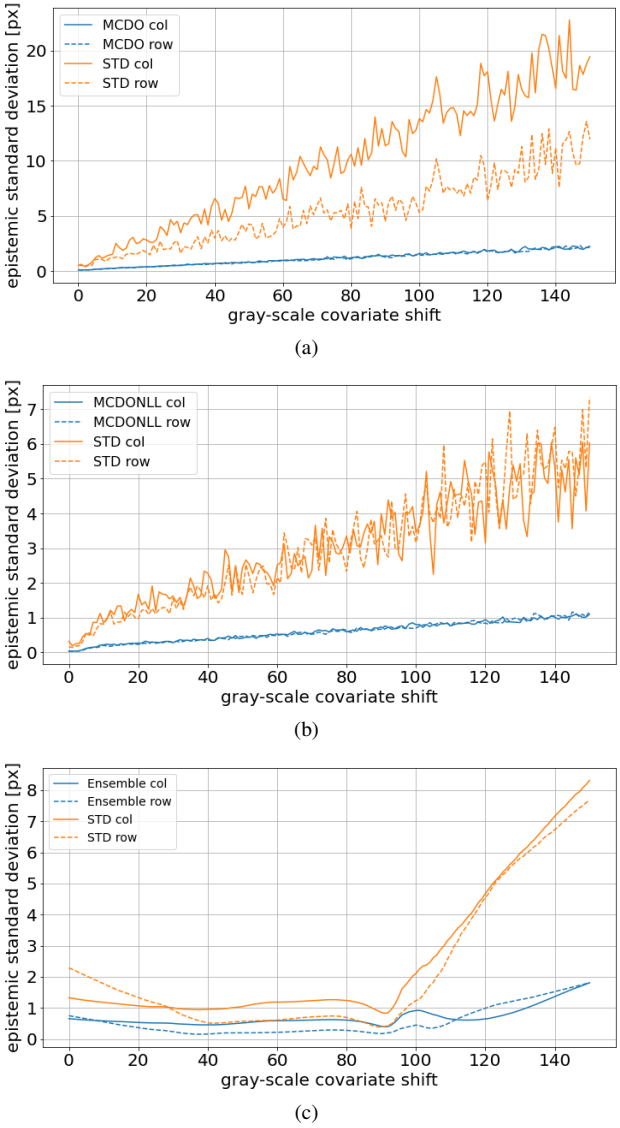
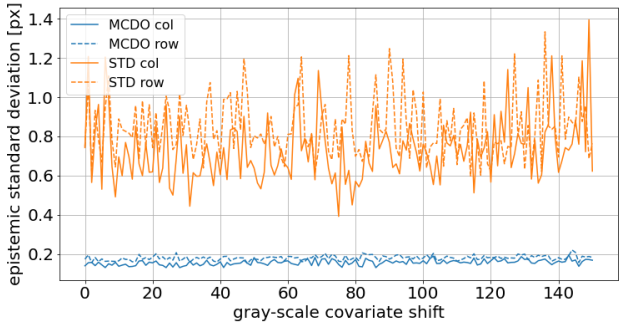


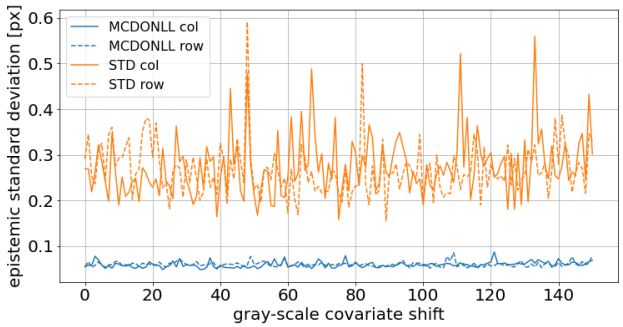
Figure 6. Epistemic standard deviation estimates in image columns and rows in pixel units with (a) MCDO, (b) MCDONLL, and (c) Ensemble under covariate shift in case of extrapolation.

Figure 7 shows the results of (a) MCDO, (b) MCDONLL, and (c) Ensemble trained on the covariate shift dataset in case of

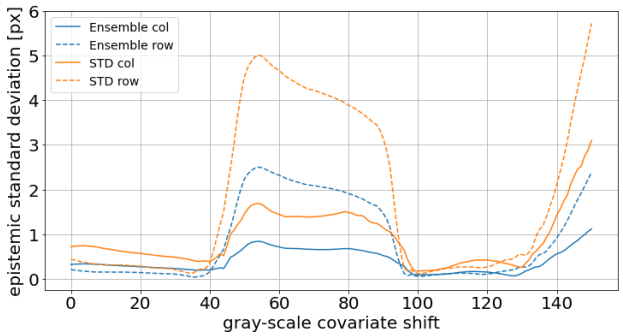
interpolation. Both MCDO and MCDONLL are able to interpolate the gap in the gray value training distribution, as it is shown by the continuous STD values. The epistemic standard deviation estimates of Ensemble are smaller than STD, however, they are in best agreement with STD. Furthermore, the uncertainties are higher for images with gray value distributions outside the training data distribution. The gray value distribution of dark training images of the covariate shift dataset ranges from around 50 to 100 and that of light images from 200 to 250 gray values. The influence of this can be seen in Figure 7(c), where the uncertainty estimates are low for covariate shift which still overlap with the training data distribution. From another perspective, the higher STD values outside of the training data distribution also suggest inferior prediction quality and a lower generalization of the trained Ensemble than in case of MCDO and MCDONLL.



(a)



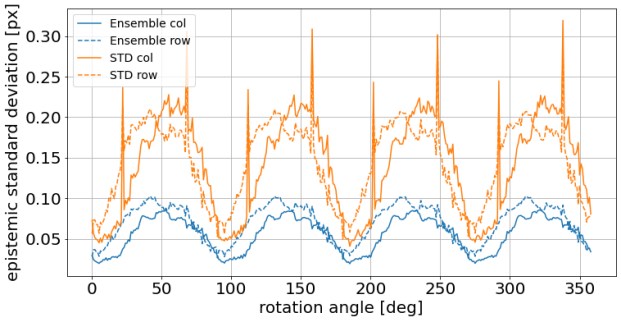
(b)



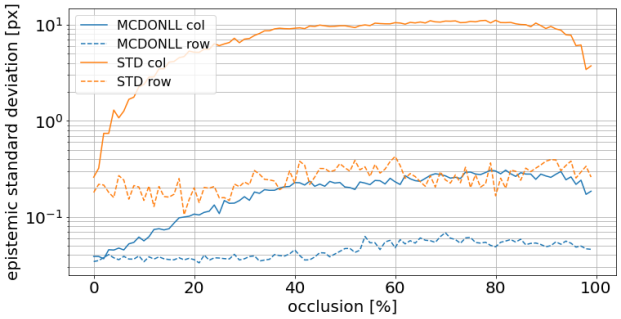
(c)

Figure 7. Epistemic standard deviation estimates in image columns and rows in pixel units with (a) MCDO, (b) MCDONLL and (c) Ensemble under covariate shift in case of interpolation.

4.1.3 Rotation and Occlusion The implemented UQ methods MCDO, MCDONLL, and Ensemble are evaluated on the rotation and occlusion dataset. The best results in terms of the agreement with the STD were obtained with Ensemble on the rotation dataset and MCDONLL on the occlusion dataset. The results are shown in Figure 8. In Figure 8(a), the systematic influence of the four symmetrical axes of the square can clearly be seen. Both estimated epistemic standard deviations and STD are highest for rotation angles of a multiple of 45 degrees and lowest for multiples of 90 degrees. Figure 8(b) shows increasing epistemic standard deviation estimates with increasing degree of occlusion of the square in the image. The uncertainty only increases for the columns, as the squares in the dataset are only occluded along the image columns.



(a)



(b)

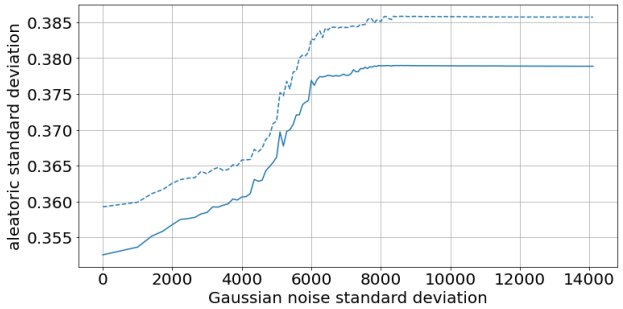
Figure 8. Epistemic standard deviation estimates and STD in image columns and rows in pixel units with (a) Ensemble on the rotation dataset and (b) MCDONLL on the occlusion dataset.

4.2 Aleatoric Uncertainty

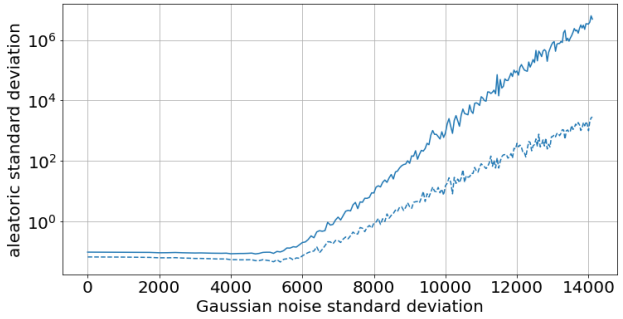
The aleatoric uncertainty estimation of ADF and NLL is evaluated on the noise dataset that consists of images with Gaussian noise of various gray values. The mean estiamtes of ten standard deviation predictions are shown in Figure 9. The aleatoric standard deviation estimates are relatively low as long as the square is still recognizable despite the noise. For higher noise levels, however, the uncertainty estimates of NLL increase exponentially, while the estimates of ADF increase only slightly.

5. CONCLUSIONS

We compared different UQ methods, namely Monte-Carlo Dropout (Gal and Ghahramani, 2016), Deep Ensembles (Lakshminarayanan et al., 2017) and the combined approach of (Kendall and Gal, 2017) for epistemic uncertainty estimation as well as (Gast and Roth, 2018) and (Kendall and Gal, 2017) for aleatoric epistemic uncertainty estimation on a simplistic dataset. All methods show high epistemic uncertainties around the borders of the training data distribution and low uncertainty estimates far outside the training data distribution. Both, models and



(a) ADF



(b) NLL

Figure 9. Aleatoric standard deviation estimates of (a) ADF and (b) NLL in pixel units on the noise dataset.

uncertainty estimates, handle interpolation under dataset shift well but not in case of extrapolation. In the case of rotated or occluded objects, the methods provide higher uncertainties in determining the position of these objects, as desired. In comparison to the absolute standard deviation estimates, Deep Ensembles achieve the best epistemic standard deviation results.

In the future, we would like to investigate the usage of UQ methods on deep learning-based 6D object pose estimation methods. In this regard, we find the Deep Ensemble approach the most promising.

REFERENCES

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., Nahavandi, S., 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76, 243-297.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D., 2016. Concrete Problems in AI Safety. arXiv preprint arXiv:1606.06565.

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight Uncertainty in Neural Networks. *International Conference on Machine Learning*, 37, 1613–1622.

Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N. D., 2009. *Dataset shift in machine learning*. MIT Press, Cambridge, Mass.

Durasov, N., Bagautdinov, T., Baque, P., Fua, P., 2021. Masksembles for uncertainty estimation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13539–13548.

Gal, Y., 2016. Uncertainty in Deep Learning. PhD thesis, University of Cambridge.

Gal, Y., Ghahramani, Z., 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *International Conference on Machine Learning*, 48, 1050–1059.

Gal, Y., Islam, R., Ghahramani, Z., 2017. Deep Bayesian active learning with image data. *International Conference on Machine Learning*, 70, 1183–1192.

Gast, J., Roth, S., 2018. Lightweight Probabilistic Deep Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3369–3378.

Hodan, T., Sundermeyer, M., Drost, B., Labbe, Y., Brachmann, E., Michel, F., Rother, C., Matas, J., 2020. BOP Challenge 2020 on 6D Object Localization. arXiv preprint arXiv:2009.07378.

Huber, M. F., 2020. Bayesian perceptron: Towards fully bayesian neural networks. *IEEE Conference on Decision and Control (CDC)*, 3179–3186.

Hüllermeier, E., Waegeman, W., 2021. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110, 457–506.

Jospin, L. V., Buntine, W. L., Boussaid, F., Laga, H., Bennamoun, M., 2022. Hands-on Bayesian Neural Networks - a Tutorial for Deep Learning Users. arXiv preprint arXiv:2007.06823.

Kendall, A., Gal, Y., 2017. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30.

Kingma, D. P., Ba, J., 2015. Adam: A method for stochastic optimization. *International Conference for Learning Representations (ICLR)*.

Lakshminarayanan, B., Pritzel, A., Blundell, C., 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *Advances in Neural Information Processing Systems*, 30.

Leibig, C., Allken, V., Ayhan, M. S., Berens, P., Wahl, S., 2017. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports*, 7(1).

Loquercio, A., Maqueda, A. I., del Blanco, C. R., Scaramuzza, D., 2018. DroNet: Learning to Fly by Driving. *IEEE Robotics and Automation Letters*, 3(2), 1088-1095.

Loquercio, A., Segu, M., Scaramuzza, D., 2020. A General Framework for Uncertainty Estimation in Deep Learning. *IEEE Robotics and Automation Letters*, 5(2), 3153-3160.

McAllister, R., Gal, Y., Kendall, A., van der Wilk, M., Shah, A., Cipolla, R., Weller, A., 2017. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. *International Joint Conference on Artificial Intelligence, IJCAI-17*, 4745–4753.

Shi, G., Zhu, Y., Tremblay, J., Birchfield, S., Ramos, F., Anandkumar, A., Zhu, Y., 2021. Fast uncertainty quantification for deep object pose estimation. *IEEE International Conference on Robotics and Automation (ICRA)*, 5200–5207.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56), 1929-1958.