Calibrating Uncertainties in Object Localization Task

Buu Phan[*] btphan@uwaterloo.ca
Rick Salay[*] rsalay@gsd.uwaterloo.ca
Krzysztof Czarnecki[*] kczarnec@gsd.uwaterloo.ca
Vahdat Abdelzad[*] vabdelza@uwaterloo.ca
Taylor Denouden[†] tadenoud@uwaterloo.ca
Sachin Vernekar[†] sverneka@uwaterloo.ca

Abstract

In many safety-critical applications such as autonomous driving and surgical robots, it is desirable to obtain prediction uncertainties from object detection modules to help support safe decision-making. Specifically, such modules need to estimate the probability of each predicted object in a given region and the confidence interval for its bounding box. While recent Bayesian deep learning methods provide a principled way to estimate this uncertainty, the estimates for the bounding boxes obtained using these methods are uncalibrated. In this paper, we address this problem for the single-object localization task by adapting an existing technique for calibrating regression models. We show, experimentally, that the resulting calibrated model obtains more reliable uncertainty estimates.

1 Introduction

In safety-critical systems such as self-driving cars, it is desirable to have an object detection system that provides accurate predictions and reliable, or well-calibrated, associated uncertainties. The uncertainties in this case come from two sources: the bounding box regressor and the object classifier. For the bounding box regressor, a $p\%$ confidence interval for each coordinate (estimated from the calibrated uncertainties) should contain the true value $p\%$ of the time. Similarly, in classification, calibration means that predictions with $p\%$ confidence are accurate $p\%$ of the time. Miscalibration, in either case, can lead to dangerous situations in autonomous driving. For instance, if the detection model indicates the 95% confidence interval that the location of a pedestrian is within the sidewalk, but it is actually a 50% confidence interval, then the vehicle may make hazardous movements.

Recent advances in Bayesian neural networks (BNNs) (Gal and Ghahramani [3], Khan et al. [8], Kendall and Gal [7]) have provided a framework for estimating uncertainties in deep neural networks (DNNs). The obtained uncertainty estimates from BNNs, however, requires calibration (Kuleshov et al. [9], Gal and Ghahramani [3]).

Recent works by Miller et al. [12] and Feng et al. [2] have shown the benefits of modeling uncertainty for detection accuracy in 2D open-set conditions and the 3D Lidar object detection task, respectively. However, neither of them has analyzed or focused on the reliability of the estimated localization uncertainties in terms of calibration. In this paper, we address this issue for the 2D single object classification and localization (SOCL) task and demonstrate its applicability on the Oxford-IIIT Pet Dataset (Parkhi et al. [13]). We focus on the localization uncertainty estimates since our experiment shows that while BNNs produce a calibrated classification uncertainty (similar to McClure and Kriegeskorte [11]'s results), the estimated localization uncertainty is not calibrated. Specifically, our contributions are: (1) we show that the estimated localization uncertainties for the bounding box

[*]Department of Electrical and Computer Engineering, University of Waterloo
[†]Department of Computer Science, University of Waterloo

Third workshop on Bayesian Deep Learning (NeurIPS 2018), Montréal, Canada.

coordinates from the BNN model are not calibrated; (2) we adapt Kuleshov et al. [9]'s method for calibrating regression models and show improvements in this setting.

The remainder of the paper is structured as follows. Section 2 gives the necessary background for the SOCL task. Section 3 describes the calibration method for localization (see Guo et al. [5] for calibration in classification). Section 4 shows experimental results demonstrating the method is effective. Finally, in Section 5 we discuss conclusions and future work.

2 Background: SOCL Task with Uncertainty Estimation

The goal of the SOCL task is to obtain a model that is able to predict the bounding box and class of an object in a given image. For an image dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, the associated labels are $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\}$, where each $\mathbf{y}_k = [\mathbf{c}_k, \mathbf{b}_k]$ consists of a one-hot encoded class $\mathbf{c}_k$ and the bounding box coordinates $\mathbf{b}_k = \{b_{1k}, b_{2k}, b_{3k}, b_{4k}\}$. We define a DNN $\mathbf{f_W}(\mathbf{x}) = \hat{\mathbf{y}}$ with weights $\mathbf{W}$ such that it predicts both the class probability and coordinates.

Incorporating Uncertainty Based on Kendall and Gal [7]'s method, we incorporate the aleatoric and epistemic uncertainties into the model by optimizing the weights $\mathbf{W}$ with heteroscedastic loss and dropout training (Srivastava et al. [15]). At test time, we sample the predictions with MC-dropout and calculate the predictive mean and uncertainties. This results in a BNN model $\hat{\mathbf{f}}_\mathbf{W}(\mathbf{x}) = [\bar{\mathbf{y}}, \bar{\boldsymbol{\sigma}}^2]$, where $\bar{\mathbf{y}} = [\bar{\mathbf{c}}, \bar{\mathbf{b}}]$ is the predictive mean and $\bar{\boldsymbol{\sigma}}^2 = \{\bar{\sigma}_1^2, \bar{\sigma}_2^2, \bar{\sigma}_3^2, \bar{\sigma}_4^2\}$ is the predictive variance (sum of the epistemic and aleatoric variance) of the coordinates, assuming that they are mutually independent [1]. Similarly to Lakshminarayanan et al. [10], we estimate the probability $p(b_i|\mathbf{x})$ as a Gaussian: $p(b_i|\mathbf{x}) = \mathcal{N}(\bar{b}_i, \bar{\sigma}_i^2)$, for $i \in 1, 2, 3, 4$.

3 Calibration Method for Estimated Localization Uncertainty

We use the cumulative distribution function (CDF) form of $p(b_i|\mathbf{x})$ in Section 2 for calibration, denoted as $P_{b_i|\mathbf{x}}(z) = \Phi(z|\bar{b}_i, \bar{\sigma}_i^2)$ where $\Phi(z|\mu, \gamma^2)$ is the CDF of Gaussian distribution $\mathcal{N}(\mu, \gamma^2)$ with mean μ and variance γ^2. For real value $z \in \mathbb{R}$, $P_{b_i|\mathbf{x}}(z)$ is the probability that the label b_i is in the $(-\infty, z]$ interval. Conversely, for a probability value q, we obtain the output z of the inverse CDF: $P^{-1}_{b_i|\mathbf{x}}(q) = z$, which means that the interval $(-\infty, z]$ is a $100q\%$ interval for b_i.

Let $\mathbb{I}_{b_i|\mathbf{x}}(q) \coloneqq \mathbb{I}[b_i \leq P^{-1}_{b_i|\mathbf{x}}(q)]$ be an indicator function that verifies the condition in the brackets. Then $P_{b_i|\mathbf{x}}(z)$ is calibrated when:

$$\mathbb{E}[\mathbb{I}_{b_i|\mathbf{x}}(q)] = q \tag{1}$$

This implies that we expect to see the $100q\%$ confidence interval to cover $100q\%$ of the label data. Calibrating a regression model means that we adjust $P_{b_i|\mathbf{x}}(z)$ such that (1) holds. To obtain a reliable uncertainty estimate, we carry out two steps: validating the uncertainty estimate and calibrating it.

Validating the Uncertainty Estimates In regression, higher estimated variance should correspond to higher expected square error. However, in practice, if the model lacks of expressiveness or does not converge, the resulting uncertainty estimates may not be valid and the calibration process will not give desired results. Thus, we use scatter plot (representing $\bar{\sigma}_i^2$ and $(b_i - \bar{b}_i)^2$) as a visualization method to validate this attribute of the uncertainty estimate before calibrating it.

Calibrating the SOCL BNN We adapt Kuleshov et al. [9] method of calibrating a BNN-based regressor for the case of bounding box estimation. Given an uncalibrated probabilistic model $\hat{\mathbf{f}}_\mathbf{W}(\mathbf{x})$ for the SOCL task with $P_{b_i|\mathbf{x}}(z) = \Phi(z|\bar{b}_i, \bar{\sigma}_i^2)$ for each coordinate and a calibration dataset $\hat{\mathbf{X}}$, $\hat{\mathbf{Y}}$, the calibration process trains a calibration model R_i whose input is $P_{b_i|\mathbf{x}}(z)$ such that $\hat{P}_{b_i|\mathbf{x}}(z) = R_i \circ P_{b_i|\mathbf{x}}(z) = R_i \circ \Phi(z|\bar{b}_i, \bar{\sigma}_i^2)$ is calibrated (see [8] for more details). After obtaining R_i, we replace $P_{b_i|\mathbf{x}}(z)$ by $\hat{P}_{b_i|\mathbf{x}}(z)$ for localization uncertainty estimation.

[1] This assumption gives an overapproximation of the bounding box extents, which is sufficient for obstacle avoidance.

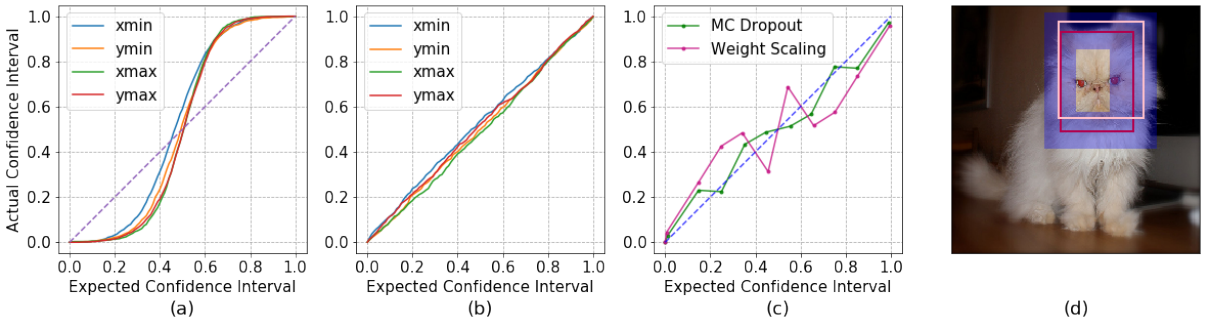

Figure 1: Reliability diagram and an example. Figure 1(a): reliability diagram for localization uncertainties before calibration. Figure 1(b): reliability diagram for localization uncertainties after calibration. Figure 1(c): reliability diagram for classification with MC-dropout and weight scaling. Figure 1(d) shows an example of bounding box localization with calibrated 95% confidence interval (blue region) centered around the mean (red). The ground truth is colored in pink

We use $[x_{min}, y_{min}, x_{max}, y_{max}]$ for encoding the coordinates in our experiments, where x_{min},y_{min} is the top left and x_{max}, y_{max} is the bottom right corner of the bounding box. To estimate the $100q\%$ confidence interval around the mean, we determine the upper bound and lower bound for each b_i by calculating $\hat{P}^{-1}_{b_i|\mathbf{x}}(r + q/2)$ and $\hat{P}^{-1}_{b_i|\mathbf{x}}(r - q/2)$ accordingly for each coordinate, where $r = \hat{P}_{b_i|\mathbf{x}}(\bar{b}_i)$. These bounds define a confidence interval as a region in which the bounding box can occur (see blue region in Figure. 1d).

4 Experiments

In the following setting, we experimentally show the miscalibration problem of localization uncertainty of our model and the improvement after applying the calibration method on the model's output. For the completeness of the task, we also show the result for classification.

We use the Oxford-IIIT Pet Dataset (Parkhi et al. [13]), which consists of 3,686 annotated images in 2 classes depicting cats and dogs, for this task. The bounding box for localization covers the face of the pet. The dataset is split into 2:0.9:0.9 ratio for training, validation/calibration and testing respectively. For the model, the VGG-16 architecture (Simonyan and Zisserman [14]) is used as a base network. The trained model obtained 94.85% classification accuracy and 14.03% localization error with 0.5 IOU threshold, based on the Imagenet evaluation method (Deng et al. [1]).We validated the uncertainty estimates (epistemic and aleatoric) and fit the calibration model R_i for each coordinate as described in Section 3.

Results Figure.1a -1c show the reliability diagram for localization and classification uncertainties on the test set. The reliability diagram shows the mapping between the expected confidence interval (from the model) and the actual one (i.e., how many labels are actually within that interval). Perfect calibration corresponds to the diagonal line in the diagram. Quantitatively, the calibration quality is evaluated by using the mean squared error (MSE) between the diagonal line and the calibration curve. Models with lower MSE are better calibrated.

Figure.1a and 1b show the reliability diagram for bounding box coordinates before and after calibration respectively. Consider the curve for x_{max} in Figure.1a, we can see that the expected 40% confidence interval corresponds to the 20% actual interval. In this case, the model has underestimated the uncertainty. On the other hand, the expected 60% confidence interval corresponds to the 80% actual interval, which implies that the model has overestimated the uncertainty in this range. After calibration, the estimated uncertainties are reliable, e.g, the expected 20% confidence interval contains approximately 20% of the true outcome. The calibration process reduces the average MSE from 2.7E-02 to 2.7E-04 for the four coordinates.

We also observed that, in Figure.1c, MC-dropout produces a calibrated classification confidence with MSE of 3.0E-03, compared to that of 1.6E-02 with the original weight scaling method for network

trained with dropout (Srivastava et al. [15]), in which we scale the weights according to the dropout rate at test time, and show the resulting softmax probabilities.

5 Conclusion and Future Work

In this paper, we consider the reliability of estimated localization uncertainty for the 2D SOCL task in terms of calibration. Our experiment shows that without calibration, the estimated localization uncertainties are misleading. We adapted an existing method for calibrating regression to the uncertainty estimates of bounding box coordinates. The result shows that the new uncertainty estimates are well-calibrated.

In future work, we would like to extend this work to the general 2D and 3D multiple object detection task and use a more complex dataset such as KITTI (Geiger et al. [4]). Furthermore, we want to address the calibration problem in the case the coordinates are not mutually independent. Finally, although the computation cost for calibration step and estimating the aleatoric uncertainty is negligible, the cost for estimating the epistemic uncertainty is high. To reduce the computation cost for this step, we want to investigate how model compression technique (Han et al. [6]) can be incorporated for speeding up the inference rate.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[2] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. *arXiv preprint arXiv:1804.05132*, 2018.

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

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[5] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017.

[6] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *International Conference on Learning Representations (ICLR)*, 2016.

[7] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017.

[8] Mohammad Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable Bayesian deep learning by weight-perturbation in Adam. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2611–2620, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL `http://proceedings.mlr.press/v80/khan18a.html`.

[9] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2796–2804, 2018.

[10] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[11] Patrick McClure and Nikolaus Kriegeskorte. Representing inferential uncertainty in deep neural networks through sampling. *arXiv preprint arXiv:1611.01639*, 2016.

[12] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. *arXiv preprint arXiv:1710.06677*, 2017.

[13] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.