Learning Uncertainty for Safety-Oriented Semantic Segmentation in Autonomous Driving

Victor Besnier, David Picard, Alexandre Briot

LEARNING UNCERTAINTY FOR SAFETY-ORIENTED SEMANTIC SEGMENTATION IN AUTONOMOUS DRIVING

Victor Besnier[1,2,3], *David Picard*[3], *Alexandre Briot*[1]

1. Valeo, Créteil, France
2. ETIS UMR8051, CY Université, ENSEA, CNRS, Cergy France
3. LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France

ABSTRACT

In this paper, we show how uncertainty estimation can be leveraged to enable safety critical image segmentation in autonomous driving, by triggering a fallback behavior if a target accuracy cannot be guaranteed. We introduce a new uncertainty measure based on disagreeing predictions as measured by a dissimilarity function. We propose to estimate this dissimilarity by training a deep neural architecture in parallel to the task-specific network. It allows this observer to be dedicated to the uncertainty estimation, and let the task-specific network make predictions. We propose to use self-supervision to train the observer, which implies that our method does not require additional training data. We show experimentally that our proposed approach is much less computationally intensive at inference time than competing methods (*e.g.*, MC Dropout), while delivering better results on safety-oriented evaluation metrics on the CamVid dataset, especially in the case of glare artifacts.

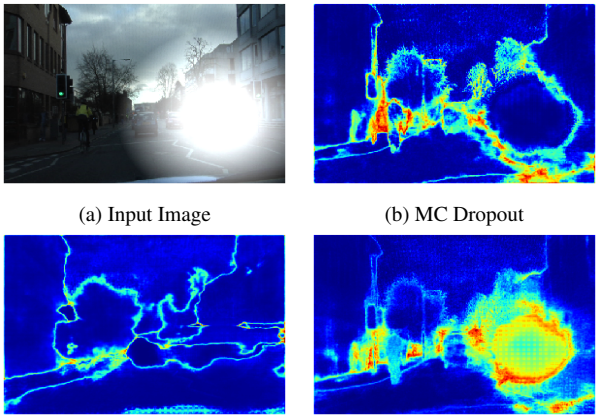
***Index Terms*—** Uncertainty, Segmentation, Autonomous Driving

1. INTRODUCTION

With the recent development of deep learning, neural networks have proven to reach or even beat human level performance at solving complex visual tasks necessary for autonomous driving such as object detection and image recognition [1, 2, 3]. However, when it comes to safety critical applications, the use of neural networks raises huge challenges still to be unlocked. Building models able to outperform humans in dealing with unsafe situations or in detecting operating conditions in which the system is not designed to function remains an open field of research. For example, although *softmax* outputs class conditional probabilities, it produces misleading high confidence outputs even in unclear situations [4].

This is all the more dramatic in autonomous driving [5] applications where the confidence of the prediction is safety critical. Demonstration of safety for AI components is a key challenge addressed by SOTIF [6] which focuses on external events (external to the system) that are not correctly handled by the system (e.g. weather conditions, user driving tasks, road users, ...). In such systems, detecting uncertain predictions is a key trigger to produce a safe behavior by interrupting the current process and starting a fallback process (*e.g.*, human intervention) instead of risking a wrong behavior.

In this paper, we focus on predicting the uncertainty of a given semantic segmentation network in an autonomous driving context. Our main contribution is a safety oriented uncertainty estimation framework that consists of a deep neural network observer (as proposed in [7]) running in parallel to the target segmentation network.



(a) Input Image (b) MC Dropout (c) Distillation (d) Ours

Fig. 1: Uncertainty maps for a noisy image. **(a)** Image with artificial sun glare. **(b)** can detect epistemic uncertainty but not aleatoric uncertainty caused by the glare. **(c)** is fast in inference but do not capture aleatoric uncertainty. **(d)** Our proposed observer detects epistemic and aleatoric uncertainty and is as fast as distillation.

This auxiliary network is trained using self-supervision to output predictions that are similar to the target network when the target network is certain, and completely different outputs when the prediction is uncertain. The uncertainty is then measured as the dissimilarity between the target network output and the observer output and we empirically show it produces a good proxy for measuring the uncertainty to detect safety critical predictions. Thus, our framework has the following properties that highlight its relevance:

- It improves over other uncertainty estimation methods at detecting safety critical predictions;
- It is trained in a simple self-supervised fashion, meaning that no expensive annotations are required;
- It does not require retraining the target network and can work with any off-the-shelf network;
- It is fast, the processing of the observer happens in parallel to the target network, meaning it has a reduced overhead compared to other uncertainty estimation methods.
- It is able to detect uncertain predictions arising from glare artifacts.

2. RELATED WORK

Semantic segmentation is an important task in autonomous driving [8]. However, wrong predictions can lead to disastrous events. As

such a confidence score has to be computed to detect unreliable predictions at the pixel level. In that context, uncertainty estimation is safety critical vision task [9].

It is well known that the *softmax* output is an overconfident score that does not correctly estimate uncertainty [10]. Even techniques to calibrate the output score like temperature scaling [10] show limitation to detect out-of-distribution sample. On the contrary, methods estimating uncertainty as the variations among multiple predictions such as [11] or [12] have shown promising results [13].

Uncertainty from Bayesian Inference. In [12], the authors propose to use Bayesian Inference to estimate the epistemic uncertainty of a model. They use dropout during training to approximate Bayesian Inference in deep Gaussian processes. During inference, they perform multiple forward passes with dropout to obtain multiple predictions. They propose to use the entropy of the average prediction as the measure of epistemic uncertainty, and show it produces promising results. However, the computational cost is very high due to the multiple forwards and it cannot model aleatoric uncertainty. [14], [15] and [16] propose similar stochastic techniques to estimate epistemic uncertainty, with the same drawbacks.

Dropout Distillation. In order to avoid the computational cost of the multiple forwards, [17] and [18] proposes to use distillation. Distillation considers two networks, a generally large one called Teacher and a generally smaller one called Student. The Student is trained so as to mimic the Teacher. In the case of uncertainty estimation, the Student is trained to capture the variance produced by the dropout in the Teacher [17]. Although distillation removes the burden of the multiple passes, it is a much more complex problem since a single network has to solve two tasks with a single head.

Sample Free Epistemic Uncertainty Estimation. In [19], the authors propose a new framework for epistemic uncertainty without multiple forward passes by measuring the propagation of the variance caused by noise in the network. To achieve this efficiently, they propose a simplification of the propagation effect in a convolution layer followed by ReLU that limits memory requirement. [20], [21] proposes similar ideas based on density estimation networks. However, similarly to all other methods, these sample-free techniques only model epistemic uncertainty and do not consider aleatoric uncertainty.

Contrarily to the related work, our proposed framework is able to capture both epistemic and aleatoric uncertainty. It is most closely related to distillation based methods in that we also use a second network. However, it is dedicated to uncertainty estimation, which we show is crucial. It is not trained to capture the variance of the main network, but is instead trained to output a second prediction that can be compared to the main prediction to estimate uncertainty. We experimentally show that it leads to better results.

3. PROPOSED METHOD

For safety purposes, it is often better to not make any decision rather than making a decision that cannot be guaranteed. Given a trigger that detects decisions that cannot be guaranteed, a safe system could then stop its current process and start a fallback process (*e.g.*, human intervention). Such a trigger divides the predictions $\hat{y}$ of a given neural network into 2 classes: the class of *certain* predictions associated with the class $c = +1$, for which an average error rate can be guaranteed; and the *uncertain* class, associate with the label $c = -1$, for which no average guarantee can be obtained. More formally, we have the following property:

$$\mathbb{E}_{\hat{y}|c=+1}[l(\hat{y}, y)] \leqslant \epsilon, \tag{1}$$

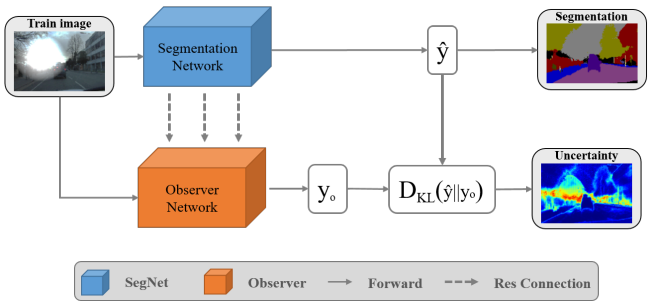


Fig. 2: Method architecture. $\hat{y}$ main prediction, y_o observer output. Our architecture is dedicated to uncertainty measurement.

with $\hat{y}|c = +1$ the distribution of *certain* predictions, y the ground truth and $l(.,.)$ the loss function of the target application.

We propose to use uncertainty estimation to obtain such safety trigger. Given a function $u(\hat{y})$ that estimates the uncertainty of a prediction $\hat{y}$, we define a safety threshold δ such that predictions under the threshold are in the *certain* set:

$$u(\hat{y}) \leq \delta \Rightarrow c = +1 \tag{2}$$

In practice, given an error rate threshold ϵ and its corresponding uncertainty threshold δ, we evaluate uncertainty functions by the recall, that is, the proportion of samples that are deemed certain.

3.1. Dissimilarity based uncertainty

To obtain $u(.)$, we propose to model the uncertainty as the dissimilarity between several predictions. Let $\hat{y}$ be the prediction of the neural network we are analyzing, and let y_a be an additional prediction without any assumption on how it is obtained (*e.g.*, additional forward of a stochastic model, ensemble, oracle, *etc.*). We propose that measuring the dissimilarity between $\hat{y}$ and y_a gives us a sense of the uncertainty regarding prediction $\hat{y}$. In the case of a regression problem, the dissimilarity can be measured as the distance between the two predictions. In this paper, we focus on softmax classification which is used in semantic segmentation and is the well studied in uncertainty estimation. It that case, $\hat{y}$ and y_a being the probability distribution over the different possible classes (softmax outputs), we propose to use the KL divergence to define the uncertainty function $u(.)$:

$$u(\hat{y}) = D_{KL}(\hat{y}||y_a). \tag{3}$$

In other words, if $\hat{y}$ and y_a produce similar outputs, the uncertainty of $\hat{y}$ is low, whereas if $\hat{y}$ and y_a produce dissimilar (or disagreeing) outputs, the uncertainty of $\hat{y}$ is high.

3.2. Learning to predict y_a

Unfortunately, it is clear that the additional prediction y_a may not be available at inference time. Our main contribution solves this problem by introducing a second predictor, the observer with output y_o which we use in place of y_a. Training y_o to precisely regress the predictions y_a can be a difficult learning problem. Moreover, in case of *uncertain* predictions, it is not required that y_o perfectly matches y_a. Instead, y_o only has to be sufficiently different from $\hat{y}$ to produce a large KL divergence, just as y_a would have done. Therefore, we propose to train y_o using a self-supervised classification problem distinguishing between *certain* and *uncertain* predictions.

In practice, given a training set of pairs $(\hat{y}, y_a)$, we use the safety threshold δ to split the pairs into *certain* $c = +1$ and *uncertain*

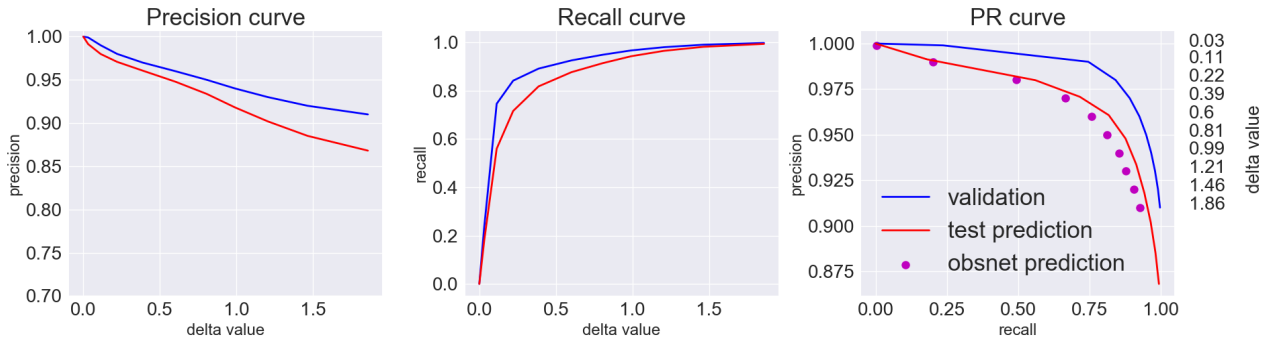


Fig. 3: Precision and Recall depending on the uncertainty threshold δ for epistemic uncertainty. Tune the hyper-parameter δ, allows to select which examples are in the *certain* prediction set (*i.e.*, $c = +1$).

$c = -1$ classes, and we train our predictor ObsNet to minimize $D_{KL}(\hat{y}||y_o)$ for $c = +1$ and maximize $D_{KL}(\hat{y}||y_o)$ for $c = -1$ by optimizing the following problem:

$$\min_{\theta} D_{KL}(\hat{y}||y_o)^c, \tag{4}$$

with θ the parameters of the observer. We argue that this objective is much easier to optimize than regressing y_a since the output y_o has many more degrees of freedom in the *uncertain* case. Instead of being force to predict the exact same class as y_a, y_o can predict any class different from the one predicted by $\hat{y}$.

Usually, uncertainty is classified into two different classes: Epistemic and Aleatoric which we both propose to estimate using different additional predictions y_a.

3.3. Epistemic Uncertainty

Epistemic uncertainty is associated with the model uncertainty. It capture the lack of knowledge about the process that generated the data. To estimate epistemic uncertainty, we propose to use MC Dropout as the additional prediction y_a. We compute the additional prediction y_a as the average of T forward passes with dropout. Note that our training setup is entirely self-supervised as the labels c are obtained using D_{KL} over forward passes of the target network only.

We show on Figure 3 what setting a specific threshold δ implies in terms of precision and recall for *certain* ($c = 1$) predictions. As we can see, the observer is perfectly able to recover the specific operating points of the uncertainty obtained by the original MC Dropout additional prediction.

3.4. Aleatoric uncertainty

The aleatoric uncertainty is associated with the natural randomness of the input signal [15]. More precisely, in this work, we focus on heteroscedastic uncertainty, which is the lack of visual features in the input data (e.g. sun glare, occlusion, ...). We propose to artificially create such cases by adding random glare noise to the input image. The prediction $\hat{y}$ is obtained by a single forward pass on the noisy image, while the additional prediction y_a is obtained by a single forward pass on the clean image.

The uncertainty function $u(\hat{y}) = D_{KL}(\hat{y}||y_a)$ then establishes whether the noise we added led to aleatoric uncertainty or not. Thanks to δ, we label the pair with either $c = +1$ or $c = -1$. As for epistemic uncertainty, we train our auxiliary network on the labeled pairs $(\hat{y}, y_a)$ using Equation 4.

3.5. Observer Architecture

To implement the uncertainty output y_o, we propose to add an observer network in parallel to the main network as shown on Figure 2. The additional network mimics the architecture of the main network. It takes the image as input as well as the activation maps of the main network as additional inputs thanks to residual connections. The uncertainty is obtained by computing the KL divergence with the main network output (*i.e.*, $D_{KL}(\hat{y}||y_o)$).

4. RESULTS

4.1. Datasets, Metrics and Baselines

In this section, we evaluate our method on CamVid [22], a dataset for road image segmentation. To compare the results, we adopt safety oriented metrics. We measure the recall when the precision is equal to 95% (e.g. R@P=0.95). Better uncertainty measures ought to achieve higher recall. Moreover, we propose a "safety trigger rate metrics" which is the percentage of images in the dataset with a coverage of certain prediction over a threshold. For instance "Trigger 75%" is the percentage of images in the test set where the coverage of safe prediction is above 75%. And we also report the area under the curve (AuPR) which is a threshold independent metric. We compare several methods:

- **Softmax [23]**: One minus the maximum of the prediction.
- **Void Class (VC)[24]**: Void/unknown class prediction for segmentation.
- **MC Dropout [25]**: We consider this as baseline. We use $T = 50$ and $T = 2$ forward passes.
- **MCDA[26]**: Data augmentation such as geometric and color transformations added during inference time to capture aleatoric uncertainty.
- **Distillation [17]**: We propose two variants of the distillation: supervised (*i.e.*, using Teacher outputs and cross-entropy with ground truth for training) and unsupervised (*i.e.*, the student only regress the teacher output).
- **Observer**: Our proposed method, with KL divergence based uncertainty training. We use two oracle: MC Dropout (self-sup) and Ground Truth (GT). The Ground Truth variant uses ground truth labels as additional prediction y_a and is used to measure the influence of having a self supervised setup.

For all our segmentation experiments we use a Bayesian SegNet [27], [28] with dropout as the main network. Therefore, our ObsNet follows the same architecture as this SegNet.

For each mini-batch, we compute 50 forward passes with dropout to compute y_a for epistemic uncertainty. For aleatoric

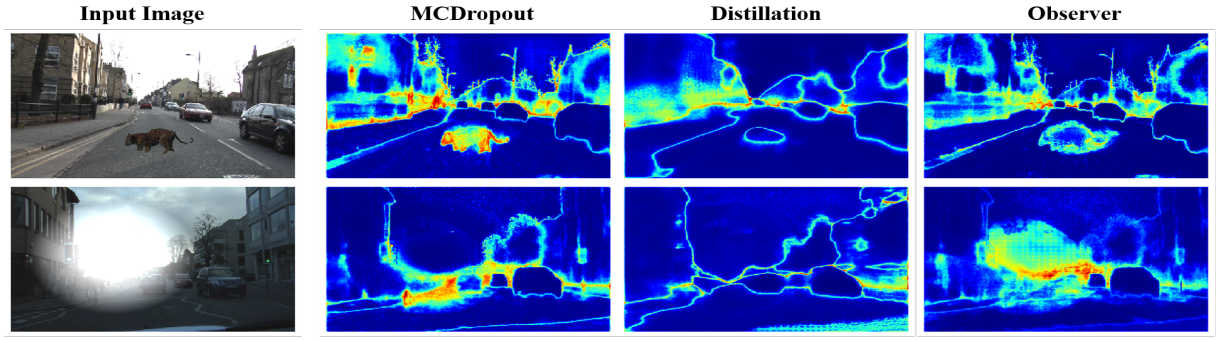


Fig. 4: Uncertainty map. First line is with an OOD animal and in the image and second one is with glare. The observer is not only close to MC Dropout for epistemic detection, but is also capable of detect aleatoric uncertainty, while distillation is perform badly.

Method	R@P=0.95	AuPR	Trigger 75%
MCDropout T50	88.2	**97.9**	**81.5**
MCDropout T2	85.9	97.5	76.4
Softmax	81.0	96.5	67.4
Void Class	64.1	95.2	39.9
Distill supervised	65.9	95.3	31.8
Distill self-sup	68.6	96.0	34.3
Observer self-sup **(ours)**	87.3	97.7	76.8
Observer GT **(ours)**	**89.3**	**97.9**	**81.5**

Table 1: Evaluation of epistemic uncertainty, best method in bold.

uncertainty, we add noise to every input image to obtain $\hat{y}$ and use the noiseless image to calculate y_a. The final loss is the sum of the aleatoric uncertainty loss and epistemic loss. The observer is trained with SGD with momentum and weight decay, by minimizing the loss Equation 4. It is trained for 50 epochs and we keep the best performing network on the validation set.

To train distillation for aleatoric uncertainty estimation, we change the training set up to be fair with our method. When the student gets an noisy image, it is train to output the same prediction as the teacher given a de-noised image.

4.2. Epistemic uncertainty

We first report results on Table 1. Our method with GT performs best or similar to MC Dropout in all the safety metrics. Our self-supervised variant method is better than MC Dropout at comparable computational cost[1]. Distillation alone offers low performances, which is due to the complexity of regressing the exact MC Dropout outputs and difficulty to capture uncertainty and class prediction.

4.3. Aleatoric uncertainty

We added glare in the image: a very important increase of brightness in an ellipse of random size and coordinates on the image. As we can see on Table 2, MC Dropout suffers dramatic failures and is unable to obtain high recall for the glare. This is to be expected since MC Dropout is not designed to capture aleatoric uncertainty. MCDA performs the best on glare among set-up without additional network. As with epistemic uncertainty, distillation does not succeed in producing good results. In contrast, our observer obtains significantly better results in all the considered cases. Overall, ours framework significantly outperforms MC Dropout and data-augmentation

[1]Our framework is equivalent to MC Dropout T2 and 20 times faster than T50 on a GeForce RTX 2080 Ti.

Method	Train	Test	R@P	AuPR	Trigger
MC Dropout T50	-	glare	0.1	83.9	18.9
MC Dropout T2	-	glare	0.0	83.7	15.5
Softmax	-	glare	1.0	90.5	17.2
Void Class	-	glare	0.3	82.7	8.2
MCDA T50	-	glare	44.7	91.7	14.3
Distill supervised	glare	glare	0.0	83.3	2.2
Distill supervised	patch	glare	0.3	85.7	4.1
Distill self-sup	glare	glare	1.6	84.1	3.0
Distill self-sup	patch	glare	1.9	85.5	4.3
Obs self-sup **(ours)**	glare	glare	68.4	95.3	23.2
Obs GT **(ours)**	glare	glare	**79.4**	**96.6**	**39.2**
Obs self-sup **(ours)**	patch	glare	46.7	92.3	15.1
Obs GT **(ours)**	patch	glare	54.5	93.8	26.2

Table 2: Evaluation of aleatoric uncertainty.

based method for aleatoric uncertainty while being much less computationally expensive.

To evaluate the generalization capabilities of our method, we train on patch noise (*i.e.*, random patch on the images are added during training) and evaluate on glare (bottom of Table 2). As we can see, although training on a different noise decreases the performance compared to the full training, we still outperforms MC Dropout, distillation and MCDA by a large margin. This shows the capacity of the observer to generalize the unseen noises.

We show qualitative uncertainty maps of Figure 4. MC Dropout and ours output very similar maps for epistemic uncertainty, while outperform others methods on noisy images.

5. CONCLUSION

In this paper, we have present a simple and effective method to estimate uncertainty. We introduce a safety oriented context where the uncertainty is used to trigger a safety signal when a given error rate cannot be met. Contrarily to ensemble methods, our framework requires a single forward pass making it computationally efficient. Contrarily to distillation based methods,our observer relies on self-supervised classes and is much more effective. With experiments on CamVid, we show that our method obtains improved results compared to competing approaches.

6. REFERENCES

[1] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He, "Aggregated residual transformations for deep neural networks," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1492–1500, 2017.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv*, 2017.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick, "Mask r-cnn," *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[4] Marcin Możejko, Mateusz Susik, and Rafał Karczewski, "Inhibited softmax for uncertainty estimation in neural networks," *arXiv*, 2018.

[5] Rhiannon Michelmore, Marta Z. Kwiatkowska, and Yarin Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," *ArXiv*, vol. abs/1811.06817, 2018.

[6] ISO/PAS 21448, "Road vehicles — safety of the intended functionality," *ISO/PAS*, 2019.

[7] SAFAD, "Safety first for automated driving," *SAFAD*, 2019.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] Rowan McAllister, Yarin Gal, Alex Kendall, Mark van der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller, "Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 4745–4753.

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger, "On calibration of modern neural networks," *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[11] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in Neural Information Processing Systems 30*, 2017.

[12] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*, 2016.

[13] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D Sculley, Joshua Dillon, Jie Ren, and Zachary Nado, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," *Advances in Neural Information Processing Systems*, 2019.

[14] Mattias Teye, Hossein Azizpour, and Kevin Smith, "Bayesian uncertainty estimation for batch normalized deep networks," *arXiv*, 2018.

[15] Alex Kendall and Yarin Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

[16] Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse, "Noisy natural gradient as variational inference," *arXiv preprint arXiv:1712.02390*, 2017.

[17] Corina Gurau, Alex Bewley, and Ingmar Posner, "Dropout distillation for efficiently estimating model confidence," *arXiv*, 2018.

[18] Andrey Malinin, Bruno Mlodozeniec, and Mark Gales, "Ensemble distribution distillation," *International Conference on Learning Representations*, 2020.

[19] Janis Postels, Francesco Ferroni, Huseyin Coskun, Nassir Navab, and Federico Tombari, "Sampling-free epistemic uncertainty estimation using approximated variance propagation," *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[20] Sungjoon Choi, Kyungjae Lee, Sungbin Lim, and Songhwai Oh, "Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[21] Michael Truong Le, Frederik Diehl, Thomas Brunner, and Alois Knol, "Uncertainty estimation for deep neural object detectors in safety-critical applications," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[22] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, 2008.

[23] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song, "A benchmark for anomaly segmentation," *ArXiv*, 2019.

[24] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena, "The fishyscapes benchmark: Measuring blind spots in semantic segmentation," *arXiv preprint arXiv:1904.03215*, 2019.

[25] Yarin Gal, "Uncertainty in Deep Learning," *PhD*, 2016.

[26] Murat Seçkin Ayhan and Philipp Berens, "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks," *Medical Imaging with Deep Learning*, 2018.

[27] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[28] Alex Kendall, Vijay Badrinarayanan, , and Roberto Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.