Study on epistemic uncertainty in polyp segmentation from gastrointestinal tract images

Jort de Jong (1397885)
Eindhoven Univeristy of Technology
j.m.d.jong@student.tue.nl

***Abstract*—In medical applications, the input provided by a model needs to be interpreted by a medical expert before progressing. Modeling epistemic uncertainty accurately is crucial in preventing ill advised decisions. A model should express any uncertainty it has about the predictions it provides. This uncertainty can then be considered by medical experts when evaluating the model output. In *Evaluating and Boosting Uncertainty Quantification in Classification*, Xiaoyang Huang et al. discuss the *"lack of a unified evaluation method"* for uncertainty modeling [16]. This makes it difficult to compare methods for modeling uncertainty. In this study a number of methods, for modeling epistemic uncertainty in image segmentation, are explored and compared. This is done on the basis of the polyp segmentation task, as two independent high quality datasets are available.**

I. Introduction

Deep learning models are not all knowing. In medical applications it is especially important to highlight their limitations. Ideally, deep learning models would provide a measure of uncertainty along with the output. This uncertainty is crucial when deep learning supports medical decision making. In recent years much work has been dedicated towards modeling uncertainty. Specifically, epistemic uncertainty, when the uncertainty is not inherent to the problem. In this study a number of methods for modeling epistemic uncertainty are explored. This is done on the basis of a case study. The datasets considered relate to polyp segmentation in colonoscopy screenings. Polyps are precursors to colorectal cancer. Early detection and assessing of these polyps significantly increases the survival rate of patients diagnosed with colorectal cancer [1].

Many methods for modeling epistemic uncertainty have been proposed in recent years. There is no unified quantitative evaluation of uncertainty estimation present [16]. This makes comparing results between methods difficult. This study aims to explore which methods are approachable and effective for the task of polyp segmentation.

II. Background & Related work

There are two major different types of uncertainty in deep learning: epistemic uncertainty and aleatoric uncertainty, depicted in figure 1.

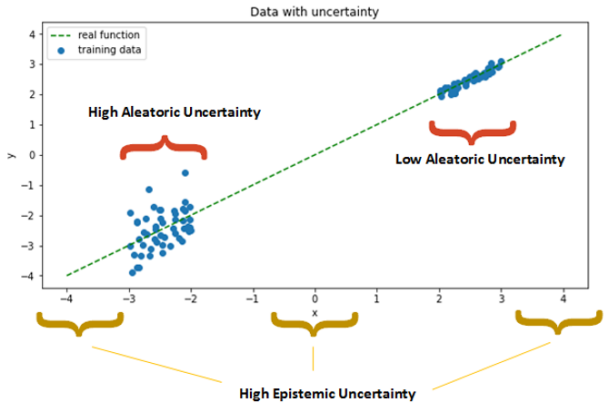


Fig. 1: Types of uncertainty [3]

A. Aleatoric uncertainty

Aleatoric uncertainty is the variability in the outcome of an experiment, due to the inherent ambiguity that exists in the data. This can be due to ambiguity in the input image (e.g. due to occlusion, shadows etc.) and in multi-annotator settings, due to the multiple ground-truth labels. This type of uncertainty cannot be explained away with more data, it is inherent to the problem. Medical application often suffer from high aleatoric uncertainty, as even experts can disagree on the ground truth [4].

B. Epistemic uncertainty

Epistemic uncertainty, sometimes referred to as model uncertainty, describes what the model does not know because training data was not appropriate. Given enough training samples, epistemic uncertainty will decrease. Epistemic uncertainty can arise in areas where there are fewer samples for training. Epistemic uncertainty can stem from noise in the data, incomplete coverage of the domain and imperfect models [5]. Like aleatoric uncertainty, epistemic uncertainty is also common in medical application. Expert annotators are often required but unavailable for large datasets. This results in limited **annotated** datasets. In medical applications

modeling epistemic uncertainty is crucial. A medical expert can disregard the model output if the epistemic uncertainty is low. Because of this, epistemic uncertainty will be the focus of this study.

C. Quantifying uncertainty

The quantification of epistemic uncertainty heavily depends on the method of modeling epistemic uncertainty. Some methods result in a distribution over a set of outputs. Here the entropy or the variance in the outputs can be used to quantity the uncertainty [15].

Generative models report the likelihood of a sample or a related metric which can be used to quantify uncertainty. Generative Adversarial Networks (GAN) consist of a generator and a discriminator. The GAN is trained to generate samples that fool the discriminator. The discriminator is trained to distinguish the generated samples from the real samples, by estimating the sample likelihood. This likelihood can be used to detect out-of-distribution samples as described in section II-H. Variational Auto-encoders learn to reconstruct inputs from a latent space. This is done by maximising a lower-bound on the likelihood of that sample. Normalizing Flows learn a map from a Gaussian distribution to the training data distribution. Normalizing Flows maximise the log-likelihood. These three generative models all report a related but different metric to quantify uncertainty.

D. Evaluation of uncertainty estimation

In *"Evaluating and Boosting Uncertainty Quantification in Classification"* [16] the authors discuss the problem of evaluating uncertainty quantification. Due to the lack of a unified quantitative evaluation. The evaluations are either qualitative [11], or highly dependent on the classification performance [17].

The authors in *"Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles"* [17] suggest that uncertainty estimation should be evaluated on two aspects, 1) calibration and 2) generalization. Calibration measures, how well the uncertainty aligns with the model accuracy on in-distribution inputs. As for generalization, it focuses on whether the model is uncertain on out-of-distribution inputs. The exact uncertainty evaluation method depends on the uncertainty quantification method.

E. Model confidence

A standard neural network allready models epistemic uncertainty, simply by interpreting the output as model confidence. The output of a neural network is not discrete but continuous. This allows the model to express some epistemic uncertainty with the prediction. The model output for a class can be interpreted as a probability. For polyp segmentation, a pixel with an output of 0.83 tells us the model is 83% certain that this pixel belongs to the polyp class. A Soft-max, or a Sigmoid activation in the case of binary classification, is used to obtain a probability distribution. The loss function is also important. The cross-entropy loss function ensures the model output aligns with the class probability [10]. Note, when the the problem has some aleatoric uncertainty the model confidence is diluted with aleatoric uncertainty.

In *"On Calibration of Modern Neural Networks"* [6] the authors discuss confidence calibration, the problem of predicting probability estimates representative of the true correctness likelihood. They state that modern neural network, unlike those from a decade ago, are poorly calibrated. Depth, width, weight decay, and Batch Normalization are important factors impacting calibration. Furthermore, the authors discus a number of techniques to calibrate model confidence.

F. Monte Carlo dropout

Model confidence is not always a reliable estimate for epistemic uncertainty. In *"Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning"* Gal et. al [11] argue, that a model can be uncertain in its predictions even with a high Soft-Max/Sigmoid output. This can happen when a sample is far from the training data. Gal et. al show that the use of dropout in neural networks can be interpreted as a Bayesian approximation of a Gaussian process, a well known probabilistic model. Dropout is used in many models in deep learning as a way to avoid over-fitting, and they show that dropout approximately integrates over the models' weights. This approach, called Monte Carlo dropout, can be used to model epistemic uncertainty.

Modeling epistemic uncertainty with Monte Carlo dropout is done by running multiple forward passes trough the model with a different dropout mask each time. Each model output can be seen as a sample, from the models' posterior distribution. The mean of these samples form our final output. The variance of these samples form our epistemic uncertainty estimation.

G. Ensemble of models

The authors in *"Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles"* [17] the authors propose to use an ensemble of deep models. The ensemble of models results in a set of outputs for a single input sample. From here, the principles are similar to Monte Carlo dropout. The epistemic uncertainty is estimated by the variance of these outputs.

H. Out-of-distribution detection

We cannot rely on model confidence or Monte Carlo dropout when a sample is far from the training data [11]. The model should express high epistemic uncertainty with these out-of-distribution samples. As said, model confidence is unable to do so reliably [11]. Some generative models can be effective in detecting out-of-distribution samples. Generative models estimate the data distribution of the training data. To detect an out-of-distribution sample, the likelihood of that

sample is estimated against the modeled data distribution. Variational Auto-encoders [13] have proven effective for this task. A Variational Auto-encoder reports the ELBO value of a given sample. This ELBO value is a lower bound on the log-likelihood. Normalizing flows are an alternative generative model. Unlike the Variational Auto-encoders, Normalizing flows output an exact likelihood (against the modeled data distribution).

In either generative model, the reported likelihood can be used to indicate epistemic uncertainty. Given a sample with a high reported likelihood by the generative model. Any discriminative model, trained on the same dataset as the generative model, should perform as expected on this sample. The output of a sample with low reported likelihood should coincide with a low epistemic uncertainty. For this sample, the discriminative model cannot be expected to perform accurately.

I. Do Deep Generative Models Know What They Don't Know?

Recently in *"Do Deep Generative Models Know What They Don't Know?"* [7] the authors, Deepmind, warn us against using generative models for out for out-of-distribution detection. The paper shows that the density learned by flow-based models, VAEs, and PixelCNNs cannot distinguish images of common objects such as dogs, trucks, and horses (i.e. CIFAR-10) from those of house numbers (i.e. SVHN), assigning a higher likelihood to the latter when the model is trained on the former. They find evidence of this phenomenon when pairing several popular image data sets: FashionMNIST vs MNIST, CelebA vs SVHN, ImageNet vs CIFAR-10 / CIFAR-100 / SVHN. These results should be taken into account whenever discussing generative models for out-of-distribution detection.

There are attempts at mitigating the problems mentioned in [7]. In *"Deep Anomaly Detection With outlier Exposure"* [9] the authors propose Outlier Exposure. With Outlier Exposure, anomaly detectors are trained against an auxiliary dataset of outliers. In *"WAIC, but Why? Generative Ensembles for Robust Anomaly Detection"* [8], the authors make use of generative ensembles for robust anomaly detection.

III. Method & Materials

The methods and materials used in the study are discussed below.

A. Kvasir-SEG dataset

The Kvasir dataset [1] comprises 8000 gastrointestinal (GI) tract images, each class consisting of 1000 images. These images were collected and verified by experienced gastroenterologist. The Kvasir-SEG dataset contains 1000 polyp images from the Kvasir Dataset V2. The resolution of the images contained in Kvasir-SEG varies from 332x487 to 1920x1072 pixels. For the Kvasir-SEG dataset the Kvasir images have been annotated with segmentations. A sample from the Kvasir-SEG dataset can be seen in figure 12. For this study, a test and validation set of 100 images each are split off. This leaves 800 images for the training set.

B. CVC-Clinic dataset

Like the Kvasir-SEG dataset, the CVC-Clinic dataset [2] comprises of colonoscopy images containing polyps. A sample from the CVC-Clinic dataset can be seen in figure 13. The images in the CVC-Clinic dataset are very similar to the images in the Kvasir-SEG dataset. Note, in appendix VIII-A the borders are different between the two samples. This is indicative of the entire dataset. The CVC-Clinic dataset serves as a different source of what should be the same type of images as the Kvasir-SEG dataset. Again, a test and validation set of 100 images each have been split off.

A CVC-Clinic noise dataset has been created by adding random Gaussian noise to images of the CVC-Clinic validation set. The Gaussian noise, with 0 mean and a standard deviation of 0.08, results in images that can be considered out-of-distribution. In figure 14 a sample from the CVC-Clinic noise set can be seen.

C. Evaluating uncertainty estimation

As discussed in section II-D, evaluating uncertainty estimation is an open problem. In this study, the segmentation model accuracy will form the *ground truth*. Segmentation accuracy is described in section III-E. If the segmentation model has low accuracy, the epistemic uncertainty estimation should be high. A high segmentation model accuracy should coincide with a low epistemic uncertainty.

Now the question remains, on which samples should the epistemic uncertainty estimation be evaluated? As discussed in section II-D, two aspects need to be evaluated, 1) calibration and 2) generalization [17]. The Kvasir-SEG test set will be used to evaluate aspect 1), calibration. The CVC-Clinic test set will tell us how the uncertainty estimation generalizes to a similar dataset. With the CVC-Clinic noise set we evaluate how the model generalizes to significant out-of-distribution samples.

D. Segmentation model

DeepLabV3 has proven effective on common segmentation tasks. The DeepLabV3 with a MobileNetV3 backend is available as a PyTorch model and will be used in this study. The DeepLabV3 model comes with a single Dropout layer, a second Dropout layer is added before the final layer. This second Dropout layer will benefit Monte Carlo dropout. Training of the segmentation model is described in detail in appendix VIII-B.

E. Segmentation model evaluation

The Dice coefficient, equation 1, is used to evaluate segmentation model performance as suggested by the Kvasir-SEG dataset authors [1].

$$\text{Dice coefficient(A, B)} = \frac{2 * |A \cap B|}{|A| + |B|} \tag{1}$$

F. Monte Carlo dropout

During inference, a sample is passed through the model several times. On each pass a different dropout mask is is applied. To derive the uncertainty for pixel x_i, T inference passes are done. f^{d_t} represents the model with dropout mask d_t. This leaves us with a set of model outputs, $\{f^{d_t}(x)|0 \leq t \leq T\}$. The final model output is given by the mean of the model's posterior distribution, as in equation 2. The epistemic uncertainty is estimated by the variance of the model's posterior distribution, as in equation 3 [12]. In this study, T is set to 100.

$$\text{Predictive posterior mean: } = p = \frac{1}{T}\sum_{t=0}^{T} f^{d_t}(x) \tag{2}$$

$$\text{Uncertainty: } = c = \frac{1}{T}\sum_{t=0}^{T} [f^{d_t}(x) - p]^2 \tag{3}$$

G. Generative model

In this study, a Variational Auto-encoder is used for out-of-distribution detection. In appendix VIII-C the model architecture and training method are discussed in detail. Note that the Variational Auto-encoder is trained on the same dataset as the segmentation model.

Variational Auto-encoders output an ELBO value. This is a lower bound of the log likelihood. The ELBO value will be used to detect out-of-distribution samples, i.e., samples that are significantly different from the training data.

As discussed in the previous section, Normalizing Flows are better suited for the task of out-of-distribution detection. Initial experiments were done with the Glow Normalizing Flow architecture [14]. However, the probabilities reported by the Glow model were outside the expected range ($[0 - 1]$). This study moved on with Variational Auto-encoders instead of fixing the implementation errors of the GLOW model.

IV. Experiments

A number of experiments and their results are described in this section.

A. Segmentation model confidence

The segmentation model is trained on the Kvasir-SEG dataset. The training method is described in appendix VIII-B. The model achieves a dice coefficient of 79.49% on the Kvasir-SEG test set, a dice coefficient of 66.35% on the CVC-Clinic test set and a dice coefficient of 28.29% on the CVC-Clinic noise set.

A final Sigmoid activation, together with the binary cross-entropy loss, allow us to interpret the model output as a probability [10]. A pixel with an output of 0.83 tells us the model is 83% certain that this pixel belongs to the polyp class. In figure 2 we can see the uncertainty in the model prediction. Figure 2 b, the label, is a binary image. The model prediction, figure 2 c, is not binary. The gray pixels represent areas where the model is uncertain.

We can interpret the pixel probability as epistemic uncertainty. Let us now investigate whether this probability actually is an accurate estimation of epistemic uncertainty. In figure 3 this relation is plotted for the Kvasir-SEG test set. Here the segmentation model has produced an output for each of the 100 test images. For every pixel the model has outputted a confidence score. These confidence scores are binned into 10 equally sized ranges. For each range we compute the probability of pixels belonging to the polyp class. We can see that the model output can be interpreted as a probability. Model confidence calibration [6] can be used to correct for misalignments in figure 3. In this case, no model confidence calibration is needed.

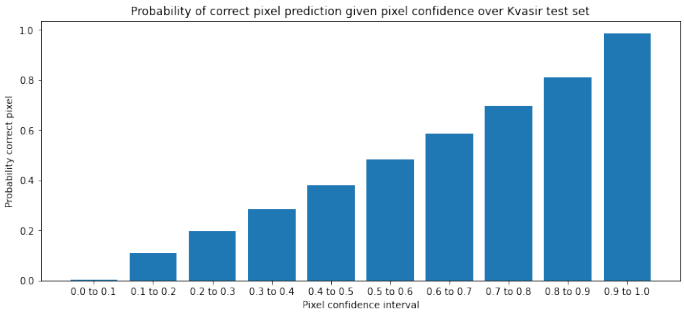


Fig. 3: Probability of correct pixel prediction given pixel confidence over the Kvasir-SEG test set

Now let us investigate whether this relation, between model uncertainty and the ground truth, holds for the CVC-Clinic test set as well. As seen in figure 4 the model output correlates with the ground truth. The relation is less linear. In figure 5 we can see the relation between model uncertainty and the ground truth for the CVC-Clinic noise set. Again, the

(a) Image (b) Label (c) Prediction

Fig. 2: Segmentation model prediction on a Kvasir-SEG test set sample

correlation exists, but the model confidence is not an accurate estimation of the probability.

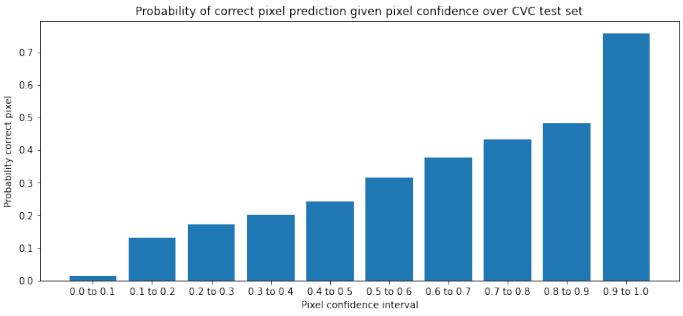


Fig. 4: Probability of correct pixel prediction given pixel confidence over the CVC-Clinic test set

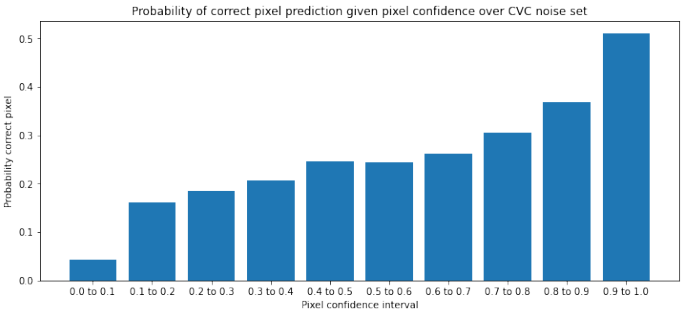


Fig. 5: Probability of correct pixel prediction given pixel confidence over the CVC-Clinic noise set

B. Monte Carlo dropout

As discussed, Gal et. al argue that a model can be uncertain in its predictions even with a high Sigmoid output. Undermining model confidence as an estimator for epistemic uncertainty. The Monte Carlo dropout method, described in section III-F, is explored to model epistemic uncertainty. In figure 6 the relation between pixel variance and the probability of a correct pixel classification over the Kvasir-SEG test set is plotted. Based on these results, a variance above 0.015 indicates high epistemic uncertainty, as the segmentation model accuracy drops to 60%.

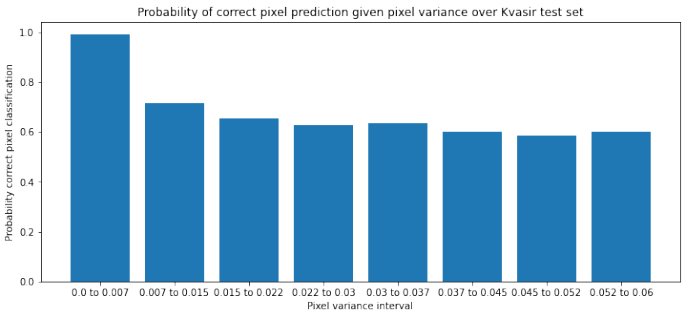
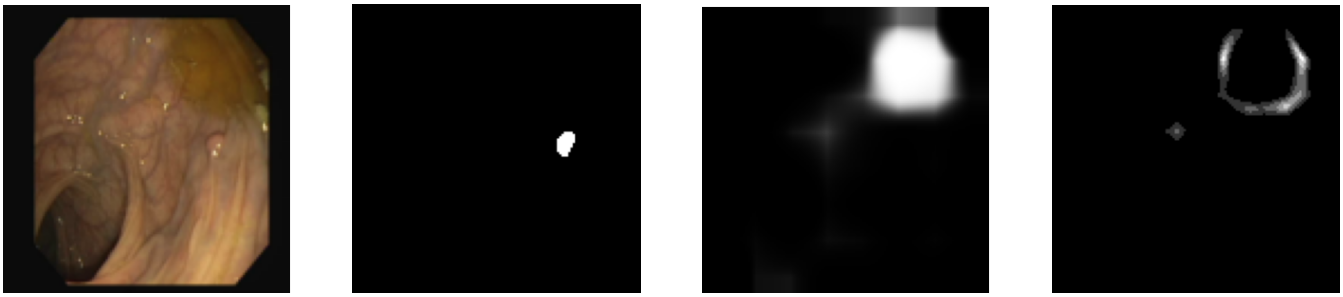


Fig. 6: Probability of correct pixel prediction given pixel variance over the Kvasir-SEG test set

In figure 7 we can see Monte Carlo dropout applied to a sample from the CVC-Clinic test set sample. The segmentation model fails to segment the polyp. In this case, the model should express high epistemic uncertainty for the false positive and false negative pixels. Model confidence is unable to do so, as seen in figure 7 c. Monte Carlo dropout is unable to improve on model confidence in this sample, as seen in figure 7 d. The Monte Carlo dropout variance coincides with the model confidence uncertainty.

Let us investigate this relation, between Monte Carlo dropout and model confidence, throughout the Kvasir-SEG test dataset. In figure 8 the pixel value (model confidence) and the pixel variance are plotted, the correctly classified pixels are colored green and the incorrectly classified pixels are colored red. A high pixel variance correlates with a low model confidence (pixel value close to 0.5). The correctly classified and incorrectly classified pixels are better distinguished by model confidence than by Monte Carlo dropout.



(a) Image (b) Label (c) Prediction (pixel mean) (d) Pixel variance

Fig. 7: Monte Carlo dropout prediction and variance on a CVC-Clinic test set sample

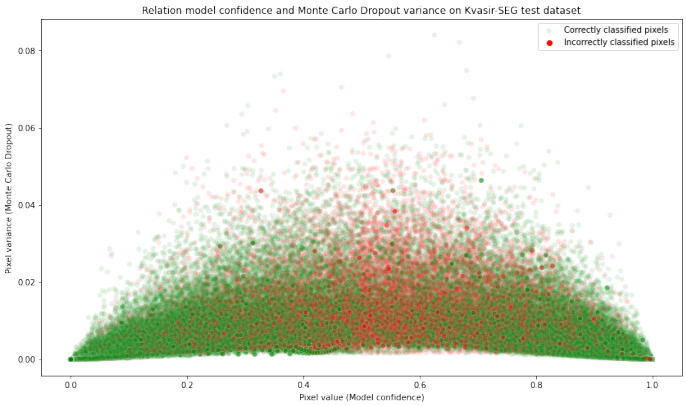
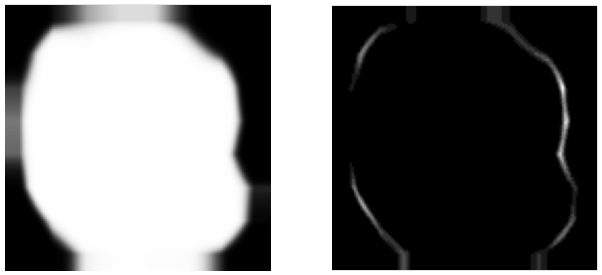


Fig. 8: Relation between model confidence and Monte Carlo dropout variance on Kvasir-SEG test dataset

Interestingly, the area with a pixel value of $\sim$0.5 and low pixel variance is quite dense. Here model confidence contradicts Monte Carlo dropout. After further investigation, these pixels are concentrated at the image borders adjacent to the polyp. Such pixels can be seen in figure 9. Here the model output is $\sim$0.5, but the pixel variance is low.



(a) Prediction (pixel mean) (b) Pixel variance

Fig. 9: Sample with contradicting between model confidence and pixel variance

C. *Out-of-distribution detection*

The VAE, as described in appendix VIII-C, is trained using the Kvasir-SEG training data. In figure 16, two sample reconstructions can be seen.

The VAE is used to compute ELBO values over the Kvasir training and test sets and the CVC-Clinic test set. The ELBO densities are plotted in figure 10. As expected the Kvasir training data has the highest estimated likelihood. This is followed by the Kvasir test data, depicted in yellow. The CVC-Clinic test data has a slightly lower overall estimated likelihood. The CVC-Clinic noise set has the lowest overall estimated likelihood. These findings align with the dice coefficient the segmentation model was able to achieve on these four datasets.

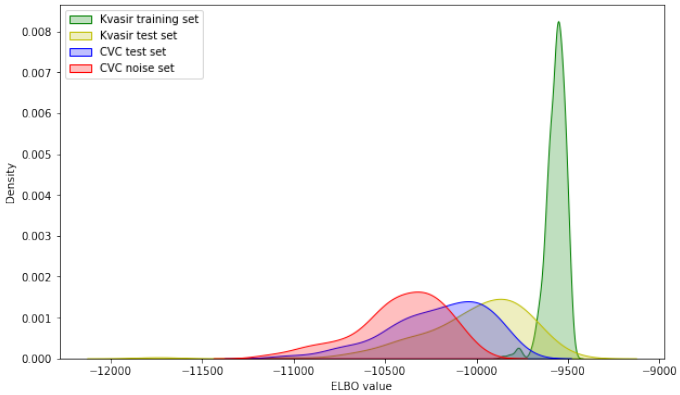


Fig. 10: ELBO densities of the Kvasir training and test sets and the CVC-Clinic test and noise sets

Let us look further into the relation between the ELBO value and the dice coefficient. The ELBO values from the Kvasir test set, the CVC-Clinic test set and the CVC-Clinic noise set are plotted against the dice coefficient in figure 11. Here the dice coefficient is provided by the segmentation model. A linear regression line has been fitted and plotted as well. As seen in the figure, the relation has a low significance.

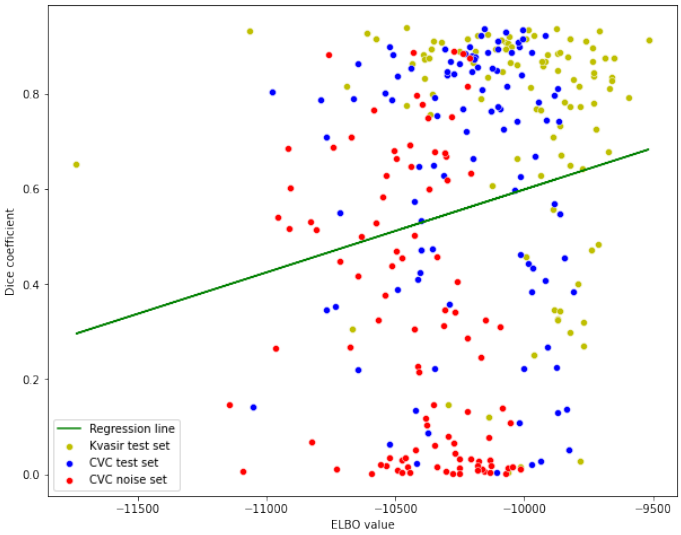


Fig. 11: Scatter plot of ELBO value and Dice coefficient of the Kvasir-SEG and CVC-Clinic test sets

V. Discussion

The segmentation model is able to model epistemic uncertainty with model confidence. In figure 3 we can see that the model output indicates the probability that the pixel belongs to the polyp class. None of the techniques presented in [6] were necessary to calibrate the model. Moving on to the CVC-Clinic dataset, the epistemic uncertainty estimation using model confidence becomes over confident. Even on the CVC-Clinic test set without noise, the model is consistently over confident.

Monte Carlo dropout models epistemic uncertainty poorly. Model confidence achieves significantly better results. In this study only two dropout layers were used in the DeepLabV3 model. More dropout layers could improve the effectiveness of Monte Carlo dropout.

Model confidence is well calibrated [17], meaning it is able to model epistemic uncertainty on in-distribution samples. Model confidence fails on the second aspect, generalization. In *"Evaluating and Boosting Uncertainty Quantification in Classification"* [16] the authors argue that classification and its confidence should be modeled separately. In section IV-C a Variational Auto-encoder is tasked with detecting out-of-distribution samples. The VAE estimates the likelihood of a sample against the training distribution reported by the ELBO value. In figure 10 the ELBO densities can be seen. Based on the results in section IV-A, we would like to see high ELBO values for the Kvasir-SEG dataset and the CVC-Clinic test set and low ELBO values for the CVC-Clinic noise set. Unfortunately there is significant overlap between the different datasets. The VAE is unable to consistently express high epistemic uncertainty for samples with a low Dice coefficient. The scatter plot, figure 11, confirms this.

There is potential for out-of-distribution detection to improve on these results. Here Variational Auto-encoders are used, Normalizing Flows allow for exact likelihood against the modeled distribution and could provide significantly better results. The amount of training data was sufficient for the segmentation model, but generative models could benefit from more data. The generative models do not require labeled data, increasing the training data with unlabeled data is possible. In section II-H a number of related works are discussed. The related works align with the results in this study, out-of-distribution detection can work but it often lacks consistency and robustness.

VI. Conclusion

Model confidence is an approachable and effective method to model epistemic uncertainty for in-distribution samples. In the case of binary segmentation, the uncertainty can be visualised with gray scale images. Model confidence does not generalise well to out-of-distribution samples. Expressing epistemic uncertainty on out-of-distribution samples is covered by out-of-distribution detection. Generative models are well suited for this task. This study as well as literature [7] highlight the difficulties, like reliability and effectiveness on limited data, in applying out-of-distribution detection.

Future work should be aimed at consistency and robustness of out-of-distribution detectors. Normalizing Flows have a lot of potential for out-of-distribution detection.

VII. Limitations

This study has limited depth and comes with some significant limitations that need to be highlighted.

The Monte Carlo dropout results were below expectations. The segmentation model contained only two dropout layers. Adding more dropout layers throughout the segmentation model could significantly improve the Monte Carlo dropout results.

In this study a Variational Auto-encoder is used to detect out-of-distribution samples. Normalizing Flows can significantly improve on the results in this study. Normalizing Flows can report the exact likelihood against the modeled distribution. The dataset, like most medical datasets, is limited by the annotation process. Training a generative model does not require labels. Extending the colonoscopy dataset with unlabeled images should improve the performance of the generative model significantly.

The methods of modeling epistemic uncertainty explored in this study is not exhaustive. Other methods, like Deep Ensembles, have been excluded from this study without substantiation.

REFERENCES

[1] The Kvasir-SEG Dataset
https://datasets.simula.no/kvasir-seg/

[2] The CVC-Clinic Dataset
https://polyp.grand-challenge.org/CVCClinicDB/

[3] Uncertainty in Deep learning
https://towardsdatascience.com/my-deep-learning-model-says-sorry-i-dont-know-the-answer-that-s-absolutely-ok-50ffa562cb0b/

[4] What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?
https://arxiv.org/pdf/1703.04977.pdf

[5] A Gentle Introduction to Uncertainty in Machine Learning
https://machinelearningmastery.com/uncertainty-in-machine-learning/

[6] On Calibration of Modern Neural Networks
http://proceedings.mlr.press/v70/guo17a/guo17a.pdf

[7] Do Deep Generative Models Know What They Don't Know?
https://arxiv.org/pdf/1810.09136.pdf

[8] WAIC, but Why? Generative Ensembles for Robust Anomaly Detection
https://arxiv.org/pdf/1810.01392.pdf

[9] Deep Anomaly Detection With outlier Exposure
https://arxiv.org/pdf/1812.04606v3.pdf

[10] What Is Cross-Entropy Loss?
https://365datascience.com/tutorials/machine-learning-tutorials/cross-entropy-loss/

[11] Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning
https://arxiv.org/pdf/1506.02142

[12] Model uncertainty in deep learning with Monte Carlo dropout in keras
https://www.depends-on-the-definition.com/model-uncertainty-in-deep-learning-with-monte-carlo-dropout/

[13] Auto-Encoding Variational Bayes
https://arxiv.org/pdf/1312.6114v10

[14] Glow: Generative Flow with Invertible 1x1 Convolutions
https://arxiv.org/pdf/1807.03039

[15] Evaluation of Uncertainty Quantification in Deep Learning
https://link.springer.com/content/pdf/10.1007%2F978-3-030-50146-4_41.pdf

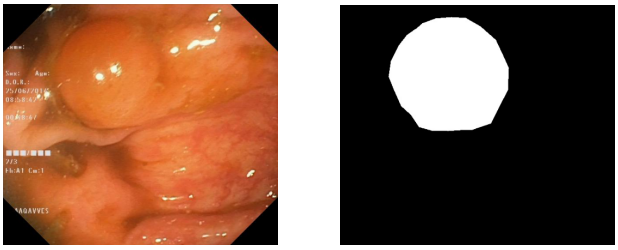
[16] Evaluating and Boosting Uncertainty Quantification in Classification
https://arxiv.org/pdf/1909.06030.pdf

[17] Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles
https://arxiv.org/pdf/1612.01474

VIII. APPENDIX

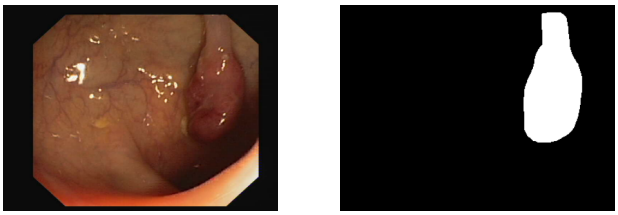
A. Dataset samples

In figure 12, 13 and 14 a sample from the Kvasir-SEG, CVC-Clinic and CVC-Clinic noise datasets are given respectively.



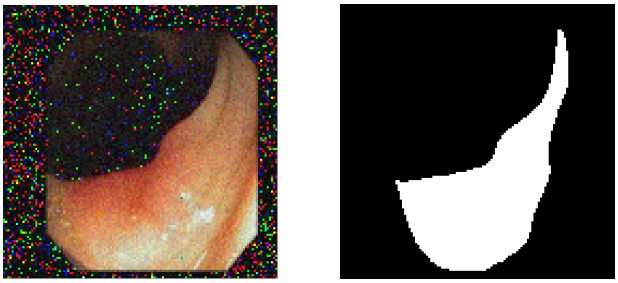
(a) Image (b) Label

Fig. 12: Sample from the Kvasir-SEG dataset



(a) Image (b) Label

Fig. 13: Sample from the CVC-Clinic dataset



(a) Image (b) Label

Fig. 14: Sample from the CVC-Clinic noise dataset

B. Segmentation model training method

The segmentation model is trained using the Kvasir-SEG dataset. The validation and testing sets are resized to 128 by 128 pixels using Bi-linear interpolation for the images and Nearest interpolation for the labels. The training images were first resized to 136 by 136 pixels. This allowed for random cropping to 128 by 128 pixels as a form of data augmentation. Color jitter, horizontal and vertical flipping were also applied at random. In [1] more dataset processing is described. A batch size of 50 was able to fit into GPU memory and was therefore used during training.

Adam was the optimizer of choice, binary cross-entropy loss was used as the loss function. Both the optimizer and

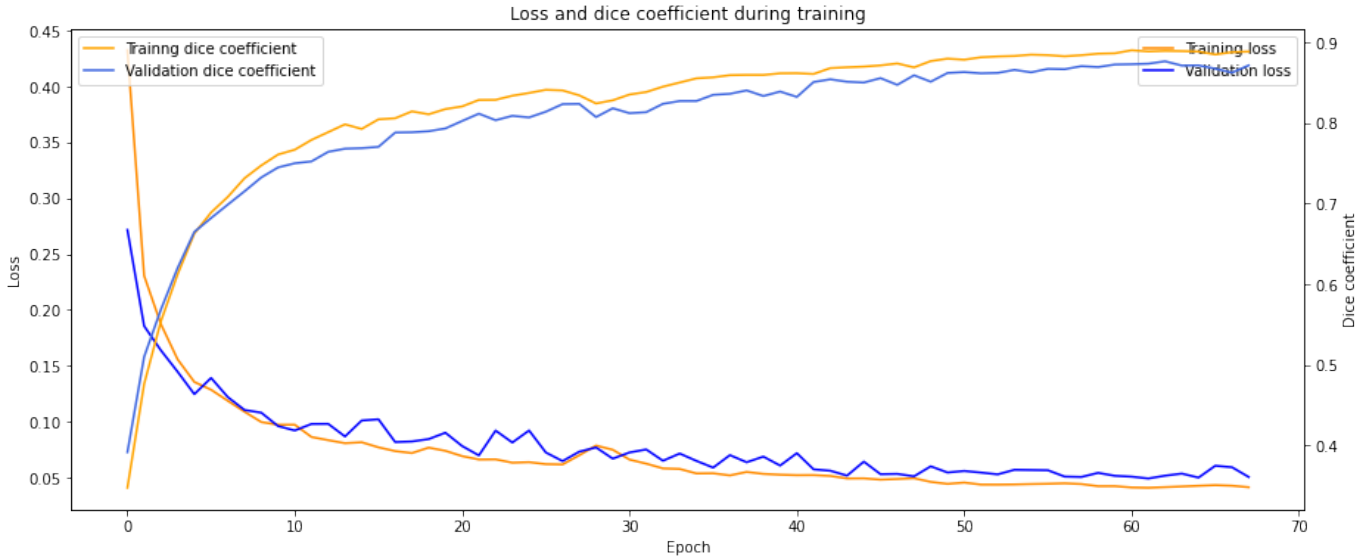


Fig. 15: Segmentation model loss and dice coefficient during training

loss function are used with the default hyper parameters. Binary cross-entropy allows us to interpret the model output as a probability distribution over each class. The segmentation model was trained until the validation dice coefficient did not increase for five epochs. In figure 15 the training and validation loss during training are plotted.

C. Variational Auto-encoder model training method

The encoder has four convolutional blocks. Each convolutional block has two series of convolutional layer, batch normalization and ReLU activation in that order. The final layer in the convolutional block is a 2 by 2 Max-Pool layer. The first convolutional layer in each block increases the number of channels. The number of channels is given as follows 3, 16, 24, 32, 40. All convolutional layers maintain spatial dimensions. After the convolutional blocks, the encoder has a linear layer that outputs 128 features. Finally the μ and the σ are produced by one linear layer each to 32 latent features.

The decoder starts of two linear layers that bring increase the dimensions from 32 to 2560 features. Four deconvolutional blocks reconstruct the original image. Transposed convolutions are followed by a batch normalization and a ReLU activation, with the exception of the final layer. The channels of the deconvolutional blocks are mirrored from the encoder. The latent distribution is Gaussian.

The Variational Auto-encoder is trained on the Kvasir-SEG dataset. All training images are resized to 128 by 128 pixels using Bi-linear interpolation. Unlike the segmentation model, the VAE is **not** trained using data augmentation. The batch size and optimizer remain the same with 50 and Adam (learning rate of 0.0014). The loss function is the negative log likelihood as proposed in [13]. The Variational Auto-encoder is trained for 400 epochs. In figure 16 a number of samples from the Kvasir-SEG test set are reconstructed.

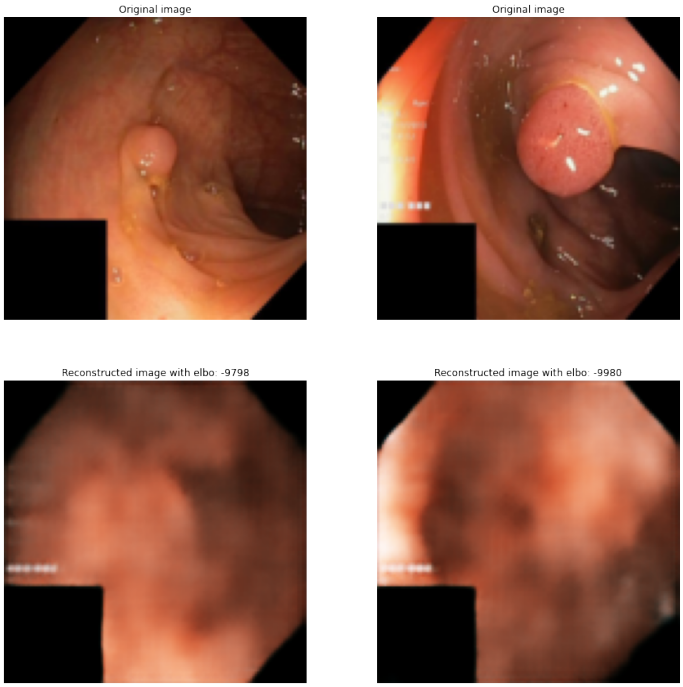


Fig. 16: Reconstruction of Kvasir-SEG test set images