ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Bayesian Deep Learning for Flood Detection

Bachelor Thesis

Timofey Shpakov

December 12, 2021

Supervisors: Prof. Dr. K. Schindler, Prof. Dr. J. D. Wegner, Dr. S. Russo

Department of Mathematics, ETH Zürich

Contents

List of Figures

List of Tables

Acknowledgements

I want to thank my supervisors, Prof. Dr. Konrad Schindler, Prof. Dr. Jan Dirk Wegner and Dr. Stefania Russo, for providing guidance and feedback throughout this thesis. Special thanks to Alexander Becker for sharing his vast knowledge on Bayesian Deep Learning and providing feedback during the final weeks of the thesis. Finally, I want to thank my family for always supporting me and for proofreading my thesis.

Abbreviations and Notation

	Abbreviations
BDL	Bayesian Deep Learning
BNN	Bayesian Neural Network
CNN	Convolutional Neural Nets
DL	Deep Learning
DNN	Deep Neural Nets
ECE	Expected Calibration Error
IoU	Intersection over Union
$mIoU$	Mean Intersection over Union
$PSPNet$	Pyramid Scene Parsing Network
SAR	Synthetic Aperture Radar
UAV	Unmanned Aerial Vehicle

	Notations
$P_{k\%}$	k-th percentile

Abstract

DNNs have set a new state-of-the-art across a various research areas. However, DNN are black box models, not giving insight on their predictive confidence, limitations and reliability. Especially in safety-critical applications, to know what your model does not know can be of utmost importance. One example is flood detection where the predictions provided by a model have to be trustworthy to efficiently plan and perform rescue operations. In this work, we evaluate a Bayesian approach on a flooded area image dataset using a combination of a deterministic Bayesian method and an ensemble method to model the aleatoric and epistemic uncertainty on the semantic segmentation of flooded area images. Furthermore, we compare the achieved model performance and calibration of the Bayesian approach with a deterministic model and show that both better performance and calibration can be achieved when taking aleatoric and epistemic uncertainty into consideration.

Chapter 1

Introduction

Floods are one of the most common natural disasters with an unpredictable nature and destructive force carrying devastating consequences to infrastructure, economy and societies. At the whim of nature, floods can affect massive areas leaving no possibility to prevent or undo the destruction following the disaster. Thus, early warning systems and a quick and effective response from rescue services make up the most essential counter-measures to floods.

Monitoring and measuring water levels and water coverage are already in use and play an integral part in early warning systems and for estimating the scale of the flood. In practice, this is currently achieved with data obtained through SRAs and UAV imagery data in combination with thresholding and statistical machine learning techniques.

This work focuses on using machine learning techniques to support decision making during rescue operations which could reduce casualties for both rescuers and rescuees by providing crucial information for rescue planning and large scale surveillance of affected areas. As an example, quick detection of flooded buildings, roads and vehicles helps in deciding which affected areas to prioritise and in developing a suitable strategy for transportation and rescue.

The application of machine learning techniques to the problem of flood detection is, however, nothing novel. Prior work has been done on the classification and segmentation of satellite and UAV images. Work on applications based on classical machine learning techniques for flood image classification has already been done and the semantic segmentation using DNN on aerial imagery data of floods is nothing novel as well. Such models, however, are often critiqued as nothing more than a black box, unable to express their confidence over their predictions. Knowing where the model is uncertain and not reliable could be particularly detrimental in the case of rescues. Rescue teams could prioritize affected areas where the predictions are more trustworthy.

Fortunately, the area of explainability and uncertainty quantification for DNNs is an active research area. The need to get an insight on confidence of DNN gave rise to a new area deep learning, namely Bayesian deep learning. In this work we make use of BNNs to tackle the problem of the prediction reliability. This work thus provides a first study of using BNNs for semantic segmentation on flood image data and how the additional information obtained from uncertainty estimates can improve predictions and help understand DNN black box models.

Chapter 2

Flood Detection

Flooding is and has been a massive concern over past decades, causing widespread damage to infrastructure and economy. Greatest damages can arise especially in the case of flash-floods where the reaction time and planning time is limited. Preventing large scale flash-floods is virtually impossible and leaving only the possibility for rescue operations during and after the flooding to reduce damages and casualties as much as possible. Thus early-warning systems and flood detection techniques are the few counter-measures one can practice in the case of flooding. In this chapter an overview over techniques for flood detection is given.

A standard conventional method incorporated in several approaches [9, 11] is thresholding which is a rudimentary image segmentation technique. The key idea is that given an image one can segment the relevant classes by colouring or grey-scale of a pixel to distinguish between classes. The method is as simple as it sounds, however, the difficulty is not in the evaluation but in constructing images which allow for thresholding to work.
An effective way to construct images in the case of flooded areas, is to use a SAR which is mounted to a satellite and actively collects data by producing a distinct signal which is yields a different response for land and water surface. The key difference between SAR and optical sensors, is that optical sensors produce data in the visible spectrum, while SAR utilizes longer wavelengths which allows data collection without disturbances by day conditions such as clouds.
In [11] the authors combine the data obtained by SAR with hydrological simulation data to improve the predictive accuracy over the conventional thresholding method while maintaining high computational speed.

Due to wide-spread adoption of machine learning, applications of statistical machine learning techniques and DL were tested on the problem of flood detection as well. [4] evaluates the performance of the classification algorithms Naive Bayes, the two decision tree bases algorithms Random Forest and J48, and a CNN, showing good performance of all algorithms on the task of water level classification. The authors concluded that Random Forest does achieve the best accuracy on the task at hand, while the CNN approach showed better overall results on the precision and recall metric [4].

With DNN and CNNs setting a new state-of-the-art in many research areas, image datasets for classification and semantic segmentation suited for training DNNs are published such as the recent FloodNet [12] which shows great performance of state-of-the-art DNN segmentation models.

While statistical machine learning methods and use of DNN have found applications in the context of flood detection, the area of Bayesian methods for flood detection is still unexplored. An interesting case study on the application of a Bayesian Network is conducted in [1].

Chapter 3

Semantic Segmentation

The goal in semantic segmentation for images is to assign a categorical label to each pixel of a given image. The output of a semantic segmentation model is typically called a segmentation mask with each pixel annotated with the predicted class. In the following sections the reader is familiarized with a few state-of-the-art architectures and basic concepts for semantic segmentation.

3.1 Models

Nowadays semantic segmentation and deep learning go hand in hand, however, finding a segmentation map does not require deep neural networks. Several methods ranging from simple computational methods such as thresholding and classical machine learning techniques such as K-Means clustering have been around for a long time. With deep learning a new state-of-the-art has been set for semantic segmentation. The characteristic components of a modern semantic segmentation model is the decoder which maps the input to a lower dimensional feature space followed by a decoder responsible for transforming the features provided by the encoder into a segmentation mask. For both components are briefly explained along with a few state-of-the-art DNN for encoding and decoding features.

Starting with encoders, they are at the heart of any computer vision task. The problem with working with image data is that we find ourselves in a very high-dimensional feature space. Each pixel in the image can be treated as separate features and with coloured images the number of features is threefold the amount compared to grey-scale images. The idea of an encoder is to find a suitable feature space of lower dimension given a high-dimensional feature space. In many application, the architecture of choice are the CNNs ResNet [5] and VGG [14].

One revolution in computer vision was due to the UNet [13] which showcases the basic structure of modern DNN for semantic segmentation. It consists of a chosen encoder, followed by the characteristic components of a semantic segmentation model the decoder. The UNet passes outputs from previous encoder layers as additional features in the decoder. The resulting architecture gives the UNet its name due to its U-shape.

Another modern segmentation model is the PSPNet [16] which is also the architecture used in the experiments as part of this work. The model is illustrated in figure 6.1. A forward pass in the model starts by passing the input image through a CNN encoder to get a feature map after the last convolutional layer. Then the characteristic PSP-module which consists of several PSP-blocks that to extract different subregion representations. They are then upsampled and all concatenated together with the feature map produced by the encoder. The final feature representation now carries both local and global context information. Finally, the final feature representation is passed through a convolutional layer to get the final per-pixel prediction [16]. By incorporating local and global context information the predictions on smaller objects can be further improved by also taking pixels on different levels of proximity into account. A good example from the PSPNet paper [16] is the confusion of the class boat and car. Having a similar shape, colour and possibly size, it is not surprising that a segmentation model might confuse them. However, if we zoom out and consider the environment we can clearly differentiate the two classes since one appears on land while the other is usually surrounded by water.

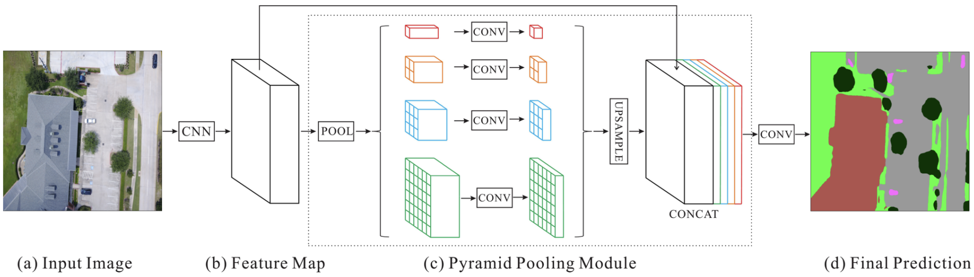

Figure 3.1: Visualization of the PSPNet architecture.

3.2 Metrics

To evaluate the quality of a semantic segmentation model we usually consider the pixel accuracy, Intersection-over-Union (IoU) and the dice coefficient.

In general, accuracy is defined as

$$\text{accuracy} = \frac{\text{Correctly classified pixels}}{\text{Total number of pixels}} \in [0,1]$$

The IoU of two sets, also called Jaccard distance, is defined as

$$\text{IoU}(X,Y) = \frac{|X \cup Y|}{|X \cap Y|} \in [0,1]$$

The dice coefficient of two sets, also called f1 score, is closely related to the IoU as it can be directly computed from the IoU and vice-versa. We define the dice coefficient as

$$D(X,Y) = \frac{2|X \cap Y|}{|X| + |Y|} \in [0,1]$$

While the pixel accuracy is the most intuitive metric, it fails to be informative in the case of highly imbalanced classes. This is the main reason for choosing IoU or Dice as the metric since both metric take the imbalance into account. In the case of semantic segmentation with C number of classes, the IoU or dice is computed for each class and finally the average of class score is taken. For IoU the formula is given by

$$\text{mIoU(X, Y)} = \frac{1}{C} \sum_{c=1}^{C} \text{IoU}_c(X,Y)$$

3.3 Losses

In order to find optimal weights with respect to segmentation, we need to define a suitable loss function. The following definitions of the loss are taken from the work [10].

A common loss for any classification task is the cross entropy loss

$$L_{CE} = -\sum_{c=1}^{C} \sum_{i=1}^{N} y_{i,c} \log \hat{y}_{i,c}$$

where $y_{i,c}$ denotes the ground truth for sample i and class c and $\hat{y}_{i,c}$ denotes the model prediction for sample i, class c and N the total number of samples or the number of samples in a batch.

Another loss used in classification is the focal loss [8]. The advantage of the focal loss compared to the cross entropy loss is that it tackles the problem of class imbalance. In essence, the loss for samples that are easily classified is reduced and for difficult samples penalized so that the model prioritizes the difficult samples. The loss is defined by

$$L_{Focal} = -\sum_{c=1}^{C}\sum_{i=1}^{N}(1-\hat{y}_{i,c})^{\gamma} y_{i,c} \log \hat{y}_{i,c}$$

where γ is the parameter that controls the loss reduction for well-classified samples. For the special case $\gamma = 0$ we have that the focal loss and cross entropy loss match.

Lastly, we consider the dice loss which is based on the dice metric. This loss function is almost exclusively used in detection and segmentation tasks as it directly optimizes the metric which is actually relevant.

$$L_{Dice} = 1 - 2\frac{\sum_{c=1}^{C}\sum_{i=1}^{N} y_{i,c}\hat{y}_{i,c} + \varepsilon}{\sum_{c=1}^{C}\sum_{i=1}^{N}(y_{i,c} + \hat{y}_{i,c}) + \varepsilon}$$

where ε denotes a small constant which is added for numerical stability.

Chapter 4

Uncertainty Estimation

In this chapter an overview is given on the current state of BDL, calibration of modern neural nets and uncertainty modelling. The discussed examples are primarily from the area of classification due to semantic segment being the focus of this work.

4.1 Calibration

Even before using Bayesian approaches, it was possible to reason about the predictive confidence of the model. In a multi-class classification setting with C labels, a model predicts a score for each class. Usually, the scores are then normalized to obtain a categorical distribution over the C labels which is obtained by applying the softmax function defined as

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \qquad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^C$$

on the scores. The obtained probability distribution is often interpreted as the model confidence.

With modern neural nets which can have from thousands to millions of parameters and numerous layers, the softmax scores are in general not reliable for indicating confidence as recent research has shown. DNN tend to be overconfident [3] in the label it predicts meaning that there is a mismatch between model performance and its self-estimation. To study the calibration of a classification model, we make use of reliability diagrams and a metric called Expected Calibration Error. In the setting of classification the softmax scores are partitioned in M confidence levels. We denote the set of predictions whose softmax score falls into the interval $I_m = (\frac{m-1}{M}, \frac{m}{M}]$ by B_m. Given the samples in a bin B_m, the average performance of the binned prediction is calculated. In classification we compute the accuracy

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}(\hat{y}_i = y_i)$$

where $\hat{y}_i$ and y_i denote the model prediction and the true label respectively. The average confidence of B_m is computed analogously

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

where $\hat{p}_i$ is the softmax score for prediction i. We call a model which achieves $conf(B_m) = acc(B_m)\ \forall m \in \{1, ..., M\}$ perfectly calibrated. The ECE is then given by

$$ECE = \sum_{m \in \{1,...,M\}} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$$

where n denotes the total number of predictions.

4.2 Uncertainty

By evaluating the calibration of a model, it can be seen that standard deterministic neural nets are already able to capture uncertainty in a sense. The model confidence then reflects a combined uncertainty of all uncertainty sources such as variability in the data and possibly a non-optimal choice of model architecture, however, it cannot distinguish between different types of uncertainty. There are several uncertainty sources which can be categorized in two different types of uncertainty, the aleatoric and epistemic uncertainty. Aleatoric uncertainty incorporates the error and noise in measurement systems that capture the data. In the setting of a computer vision task the measurement system is usually a camera which cannot give a perfect representation of the real world but an approximation. When talking about epistemic uncertainty, we consider the uncertainty introduced by the choice of model, errors in the training procedure such as using unsuitable data augmentation techniques.

The work [6] has already shown that while epistemic uncertainty can be eliminated with a sufficient amount of data, the data intrinsic aleatoric uncertainty cannot be reduced.

4.3 Methods for Uncertainty Estimation

While it is not possible to directly measure uncertainty, it can be effectively modelled with uncertainty quantification methods. We differentiate between four different types of methods to model uncertainty in our model and data

[2], namely deterministic methods, BNNs, ensemble methods and the drastically different approach the test time data augmentation. Not all methods are suited to model both aleatoric and epistemic uncertainty, however, we can combine techniques from each type to achieve both. For each type a quick overview is given.

Roughly, deterministic methods make use of a separate model or the extension of a model to explicitly model and quantify uncertainty. There exist countless methods which can be applied during training and after the training of the network. In this paragraph we focus on the particular approach of Kendall and Gal [6], which is used during the experiment of this work. The idea is to force a model to predict the parameters of the distribution which in case of a Gaussian distribution is the expected value μ and the variance σ^2.

$$\hat{x}_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

This is archived by simply doubling the output from the model. In the case of a Gaussian distribution we aim to maximize

$$\log \mathbb{E}_{\mathcal{N}(\hat{x}_i;\mu_i,\sigma_i^2)}[\hat{p}_{i,c}]$$

with $\hat{p}_{i,c}$ denoting the softmax score for sample i and class c. Since no analytical solution is known to the specified log likelihood, it is approximated by Monte Carlo Integration. In practice, this means that we take T samples from the Gaussian distribution

$$\hat{x}_{i,c} = \mu_i + \sigma_i \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, I)$$

and minimize the numerically-stable loss for the prediction i

$$L_i = \frac{1}{T} \sum_{t=1}^{T} \exp(\hat{x}_{i,t,c} - \log(\sum_{c'=1}^{C} \exp(x_{i,t,c'})))$$

. During inference we sample T outputs in the same fashion and average over the logit vectors $x_{i,t}$. To obtain a measure for the aleatoric uncertainty, we finally compute the entropy over the softmax scores for each class c

$$H(\hat{p}_i) = -\sum_{c=1}^{C} \hat{p}_{i,c} \log_2(\hat{p}_{i,c})$$

as proposed in [6].

With Bayesian methods we take a more probabilistic approach to find the best parameters of our model. The goal of Bayesian methods is to model the posterior distribution over the parameters of a model $p(\theta|x, y)$. During inference the model parameters are then sampled from the posterior distribution, where the sampled model is again a single deterministic model. The authors of [2] further subdivide Bayesian methods into three methods, namely Variational inference, sampling approaches and Laplace approximation. The differentiation between the three approaches is based on how the posterior distribution is inferred to approximate Bayesian inference.

In ensemble methods we derive a prediction based on several models which are called ensemble members. The ensemble can be a set of deterministic models which are optimally diverse and not converge to the same optimum. The key challenge when training ensemble is thus to introduce variety between the ensemble members. This can be achieved by using a different initialization for each network, data shuffling, bagging, boosting or even using different model architectures [2]. Ensemble methods are also the approach used in this work. During the experiment, the methods used introduce variety during are limited to data shuffle, random initialization and the randomness of data augmentation on the training data.
In the same fashion as Kendall and Gal [6], we obtain a measure for the epistemic uncertainty, the logit variance of the predictions from each model $\hat{x}_i$ is computed to measure the disagreement of the ensemble members.

Finally, test time data augmentation is to generate several samples from each test sample during inference by applying data augmentation techniques on the test sample. From the prediction on the augmented test samples we can then compute a predictive distribution to measure uncertainty. [2].

Chapter 5

Dataset

During the experiment in this work we use the recently published FloodNet dataset [12] which is a collection of high resolution aerial images of post flooded areas after the hurricane Harvey in Texas and Louisiana in August 2017. FloodNet includes data for three computer vision tasks, namely classification, semantic segmentation and visual question answering. Only the semantic segmentation of the FloodNet is discussed in this chapter, due to being the focus of this work.

Although image datasets for flooded areas already exist, FloodNet aims to eliminate caveats of other flood and post catastrophe datasets by providing high resolution and consistently sized UAV images of 4000x3000. By using UAVs to capture both the problem of high revisiting period and low spatial resolution of satellite imagery can be eliminated. Especially when the data is urgently required one cannot rely on satellite imagery, whereas UAVs can be quickly and accurately deployed.

The dataset consists of 2343 image-segmentation-mask-pairs with each pixel of the mask labeled with one of ten classes which include building-flooded, building-non-flooded, road-flooded, road-non-flooded, water, tree, vehicle, pool, grass and background. The background class is essentially everything that is not one of the first nine classes.

The authors of FloodNet do not only contribute by providing a new flooded area image dataset but also provide first benchmark of state-of-the-art DNN on the data with one being the PSPNet [16] which is the also architecture used during the experiments of this work.

Chapter 6

Experiments

In this section we discuss the setup such as architecture, loss and optimizers used during training of our model, followed by the results of the BNN compared to a deterministic DNN basline.

6.1 Setup

First, we explore the FloodNet dataset which contains a total of 2343 pixel-wise labeled images for semantic segmentation and the data is already split into train, validation and test set. The data split was performed with stratification over the labels meaning that the label frequency of each label is approximately the same over the training, validation and test set. Additionally, the data is split based on scenery since the dataset includes many consecutive snapshots. Having similar sceneries in both the train and validation set gives biased results since the model has already seen the data.

Just as in many datasets for semantic segmentation the classes are fairly imbalanced as seen in figure 6.1. With the grass class making up to 60% of all pixels and the vehicle and pool class being the most underrepresented classes in the dataset, it is sensible to apply measures to battle the imbalance. However, achieving best performance on each class is not the focus of this work and thus the implementation of balancing techniques is omitted.

Before even passing input through our model, we apply rudimentary image augmentations to the input images. Since the unprocessed images have dimensions of 4000x3000 we will quickly run into memory issues as an image of such as size does not fit into most memory. Thus we resize the images to 712x712 which allows us to do batch training by considering several images during an optimization step. To obtain even more data from the given dataset, we apply random horizontal and vertical flips, shifts, scaling and rotations.

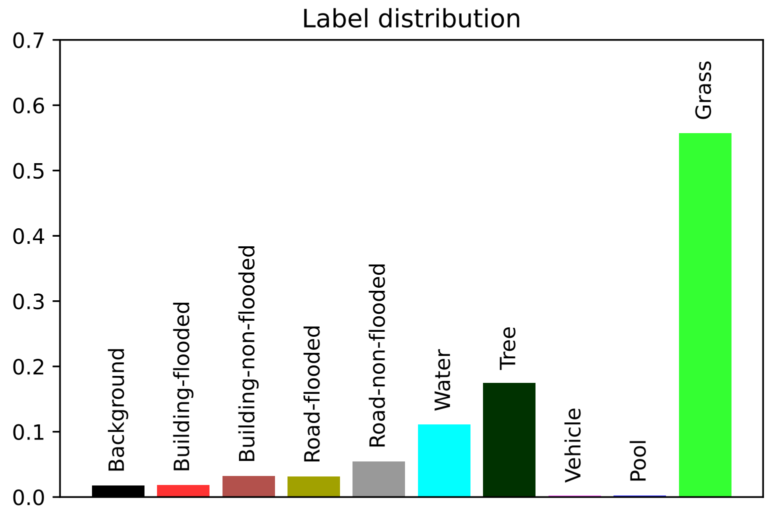

Figure 6.1: Distribution of the labels in the training set of FloodNet. The column represent the frequency pixels with the respective label. The columns are coloured with the colour used in the predicted segmentation masks as a visual aid.

Finally, we standardize the augmented images using the channel mean and standard deviation of the training data.

To understand the differences between the Non-Bayesian PSPNet, BNN and the Ensemble-BNN we carry out the experiments for the three models with the same setting.

The architecture used for the semantic segmentation model was the PSPNet with a ResNet101 without pretraining [5] as the encoder and the PSP-decoder as the decoder as outlined in section 3.1. The models were not implemented from scratch but were taken from the Segmentation Models Pytorch Python library [15] and augmented such that an untrained encoder could be used during training. The PSPNet used consists of four PSP-blocks where each block consists of an average pooling layer followed by a convolutional layer with batch normalization. Finally, we pass the output of the convolutional layer through the ReLU activation function defined a

$$ReLu(x) = \max(0, x)$$

where x denotes the output after the convolutional layer. The convolutional layer after the PSP-module is followed by batch normalization and a dropout layer with dropout probability of 0.2 which is a measure against overfitting. The rough architecture of the PSPNet decoder is outlined in section 3.1

For the BNN and ensemble of BNNs, we closely follow the approach introduced in the work of Kendall and Gal [6] and summarized in the section 4.3. The single change for the BNN compared to the Non-Bayesian model is that the output channels are doubled for predicting an output distribution. For the BNN sample the modelled aleatoric uncertainty ten times for each prediction and for the ensemble add five BNNs as our ensemble members.

Concerning the loss, in experiments with the cross-entropy, focal and dice loss, we achieved the best performance using the standard cross-entropy which does not take any imbalance into account. However, the cross-entropy loss can be attached with a weight per class to incentivise learning to correctly classify rare classes. Finally, we used the Adam optimizer [7] to minimize the cross-entropy loss with learning rate 0.0001 and weight decay 0.0001 which works as a regularizer by placing a Gaussian prior over the network weights.

We train each single model for 100 epochs each and pick the best parameters by monitoring the mIoU on the validation set.

During training we use NVIDIA GeForce GTX 1080 Ti GPU.

6.2 Results

6.2.1 Semantic Segmentation Performance

First, we evaluate the performance of the three models on the test set of the FloodNet dataset. All three models achieve a considerable improvement over the baseline in the initial experiments presented in the FloodNet paper [12]. The BNN does not surpass the performance of the Non-Bayesian baseline as seen in table 6.1, however, the mIoU of the BNN is comparable to the mIoU of the baseline while also providing modelled aleatoric uncertainty. The slight difference can be due to the randomness of the training procedure of the models. Especially interesting in the comparison of the two models, is the difference in the per-class IoU. We see an increased score on the BNN for the background class which implies the ability of BNNs to differentiate between the nine standard classes and the anomaly background class which is everything that does not belong into any of the nine other classes.

Looking at the performance of the ensemble of BNNs, the model achieves the best mIoU score and best per-class scores on most classes and comparable performance with the best achieving score of the other models. The considerable increase of more than a percentage point in mIoU in the ensemble confirms that ensemble methods do outperform and generalize better than standard Non-Bayesian models.

Table 6.1: Comparison of the baseline with the Bayesian models.

Model	Background	Building Flooded	Building Non Flooded	Road Flooded	Road Non Flooded	Water	Tree	Vehicle	Pool	Grass	mIoU
Baseline	85.16	94.48	79.69	94.29	74.16	81.11	69.87	69.80	85.58	79.02	81.32
BNN	89.84	94.89	81.98	94.93	75.06	75.08	69.80	66.96	84.36	79.40	81.23
Ensemble	**90.73**	**95.32**	**82.27**	**95.03**	**75.02**	**80.09**	**72.33**	**70.21**	**84.85**	**80.75**	**82.66**

6.2.2 Quality of Uncertainty Metric

Looking at the obtained uncertainties in figure 6.2, we can see that high aleatoric uncertainty is especially present at the boundaries between classes and for objects farther away from the UAV which was also observed by the authors in [6]. This is expected as boundaries and objects farther away are impacted the most when capturing, scaling and applying augmentations to the image. Also remarkable is that high aleatoric uncertainty has similarities to the actual prediction error which hints at a correlation between high aleatoric uncertainty and prediction error.

As for the epistemic uncertainty, we do not see the same correlation with the prediction error, however, it can be seen that it models an entirely different uncertainty than aleatoric uncertainty. Highest epistemic uncertainty is especially high inside areas of a single class and rarely found at edges. To be precise the epistemic uncertainty high, where we have high variability in the area of a single class such as colour changes and depth.

If we compare the spread of aleatoric and epistemic uncertainty, we see that per-pixel aleatoric uncertainty has a larger relative difference between most of the values compared to the per-pixel epistemic uncertainty. Noteworthy is that the epistemic uncertainty contains extreme outliers as seen in the maximal and minimal value, while most values are approximately in the same order of magnitude. This means that in most predictions less uncertainty can be attributed to epistemic uncertainty and more to the aleatoric uncertainty.

A quantitative analysis of the behaviour of the aleatoric and epistemic uncertainty can be seen in figure 6.3. We sort the pixels based on the respective uncertainty metric and remove pixels with highest uncertainty in each step. The figure shows that high aleatoric uncertainty correlates with accuracy as we see a steep increase in performance after the first 20 percentiles removed implying that high aleatoric uncertainty is a reliable indicator for misclassification. We also observe that starting from 75% of the pixels removed, we do not gain any additional increase in performance meaning that pixels with lowest aleatoric uncertainty are highly likely to be correctly classified.
For epistemic uncertainty we do not see a behaviour as in the case of aleatoric uncertainty. The epistemic uncertainty a measure of uncertainty still performs better than a random removal of the pixels.
This supports the hypothesis that aleatoric uncertainty dominates the total

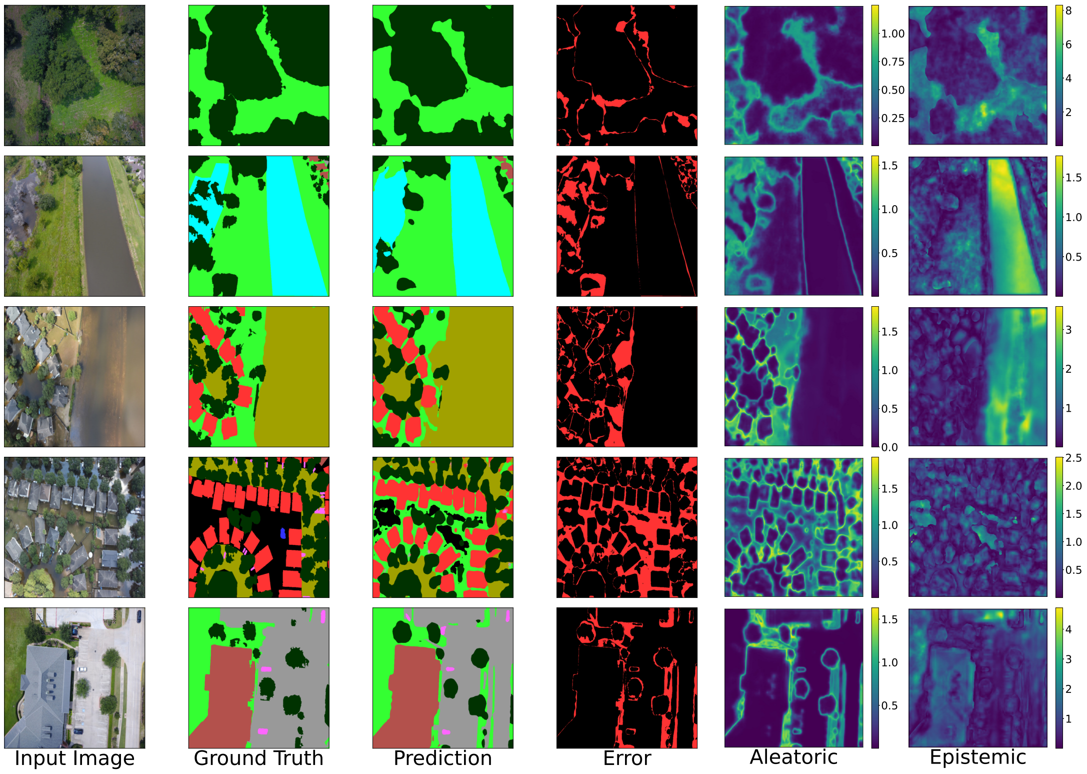

Figure 6.2: Comparison of the model prediction with the ground truth, the prediction error and illustration of the uncertainty estimates obtained through the BNN. The colouring is relative for each single image and not coloured based on the absolute value over the uncertainties in the predictions.

Table 6.2: Spread of aleatoric and epistemic uncertainty values.

Uncertainty	min	$P_{25\%}$	$P_{50\%}$	$P_{75\%}$	$P_{90\%}$	$P_{95\%}$	max
Aleatoric	0.0	0.015	0.102	0.465	0.853	1.114	2.151
Epistemic	0.00018	0.410	0.838	1.457	2.307	2.950	15.47

uncertainty during inference.

6.2.3 Calibration

Finally, we evaluate the calibration of the Non-Bayesian model, a single BNN only considering aleatoric uncertainty, an ensemble of BNN only taking epistemic uncertainty into account and the ensemble of BNN taking both uncertainty types into account. We compute the calibration and the ECE as introduced in section 4.1 with 20 bins.

We see that all the models are well-calibrated starting from a predictive

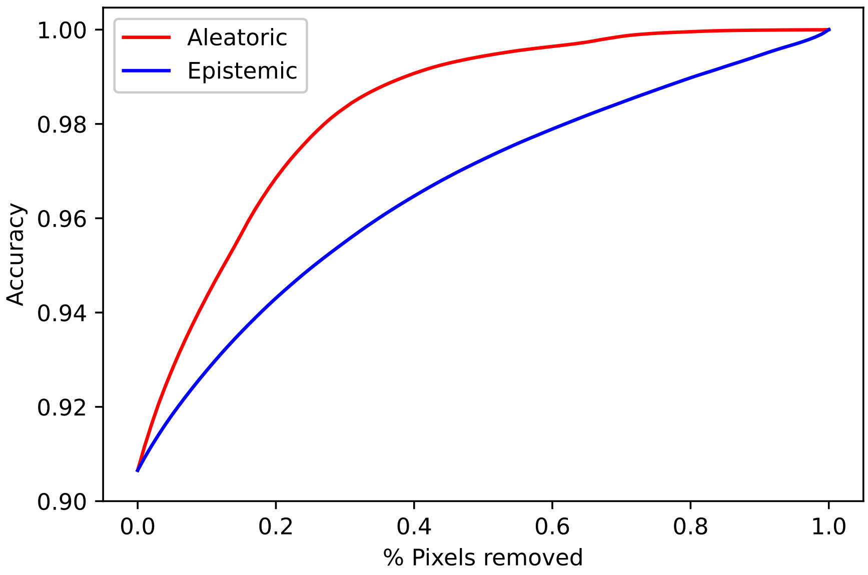

Figure 6.3: Comparison of uncertainty estimates in how removal of highly uncertain pixels improves accuracy.

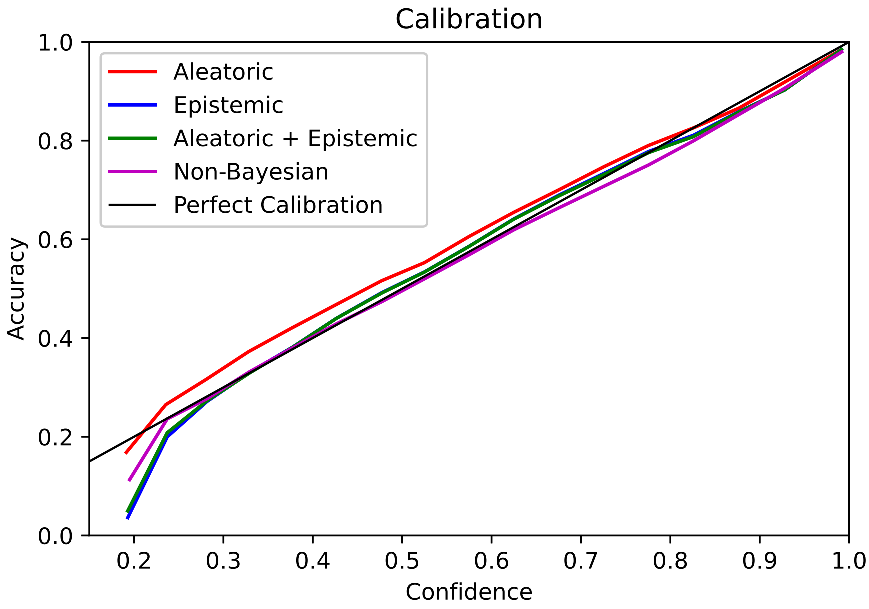

Figure 6.4: Calibration of the different models with 20 bins. A perfectly calibrated model follows the black line.

Table 6.3: Comparison of ECE of the different models with 20 bins.

Model	ECE
Non-Bayesian	0.0136
BNN	0.0114
Ensemble	0.0096
BNN Ensemble	**0.0095**

confidence of 0.25. For the lowest confidence bin of $[0.15, 0.2)$ the models underperform, however, the confidence is close to random guessing for a classification problem with ten classes. For all other confidence levels the curves closely match perfect calibration.

It is impossible to tell by eye which model is calibrated best and thus we look at the ECE in table 6.3 which hints at an improvement of calibration of the Bayesian models over the Non-Bayesian model. This implies that including aleatoric uncertainty can improve model calibration and that calibration can be further by utilizing an ensemble of BNN.

Chapter 7

Conclusion

In this work we have shown that using an ensemble of BNNs can considerably improve predictive performance and give insights on where the model fails to correctly classify samples. The knowledge of the model's limitations is especially detrimental in safety critical applications and thus gives Bayesian models an edge over Non-Bayesian models in that setting.

Another benefit of Bayesian models over Non-Bayesian models is an increase in model calibration as seen in the experiments. A higher calibration score in a model makes it more reasonable to trust the black-box predictions of the model.

Still concerning, is that the model struggles to identify classes with very few data samples such as vehicles. A remedy can be using a weighted loss with which it would be possible to prioritize learning rare or important classes. Especially in the context of flood detection and rescues, one might want to prioritise the identification of vehicles over a pool or tree class.

Furthermore, a Bayesian patched neural network architecture could be used to segment images. The benefit being that the full image is used without introducing any additional noise into the data by rescaling the image to a smaller size.

Finally, a suitable model fit for flood detection and rescue operations should be trained and tested on more flooded imagery datasets for further generalization.

Bibliography

[1] Annarita D'Addabbo, Alberto Refice, Guido Pasquariello, Francesco Lovergine, Domenico Capolongo, and Salvatore Manfreda. A bayesian network for flood detection combining sar imagery and ancillary data. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–14, 03 2016.

[2] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks, 2021.

[3] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017.

[4] Abdirahman Hashi, Abdullahi Abdirahman, Mohamed Elmi, Siti Hashi, and Octavio Rodriguez. A real-time flood detection system based on machine learning algorithms with emphasis on deep learning. *International Journal of Engineering Trends and Technology*, 69:249–256, 05 2021.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[6] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[8] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[9] S. Martinis, A. Twele, and S. Voigt. Towards operational near real-time flood detection using a split-based automatic thresholding procedure on high resolution terrasar-x data. *Natural Hazards and Earth System Sciences*, 9(2):303–314, 2009.

[10] Ty Nguyen, Tolga Ozaslan, Ian D. Miller, James Keller, Giuseppe Loianno, Camillo J. Taylor, Daniel D. Lee, Vijay Kumar, Joseph H. Harwood, and Jennifer Wozencraft. U-net for mav-based penstock inspection: an investigation of focal loss in multi-class segmentation for corrosion identification, 2018.

[11] Masato Ohki, Kosuke Yamamoto, Takeo Tadono, and Kei Yoshimura. Automated processing for flood area detection using alos-2 and hydrodynamic simulation data. *Remote Sensing*, 12(17), 2020.

[12] Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding, 2020.

[13] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[15] Pavel Yakubovskiy. Segmentation models pytorch. `https://github.com/qubvel/segmentation_models.pytorch`, 2020.

[16] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network, 2017.

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Bayesian Deep Learning for Flood Detection

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Shpakov	Timofey

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zurich, 12.12.2021	Timofey Shpakov

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.