UPPSALA
UNIVERSITET

PROJECT REPORT

Can You Trust Your Deep Neural Network?

Estimating Certainty of Deep Learning

Melanie Andersson, Sara Hedar & Andreas Isaac

Project in Computational Science: Report

February 2020

Department of Information Technology

Abstract

Deep neural networks' inability to predict their certainty is an issue affecting their usage. The aim of this study is to evaluate methods for achieving more well-calibrated models. In addition, a number of measures for estimating the certainty of a classifier are evaluated. The studied methods are temperature scaling, ensemble methods and Stochastic Weight Averaging Gaussian. Five certainty measures were implemented to evaluate these methods on three different network structures. The three measures that involve binning the predictions were shown to provide the most consistent results. The results suggest that, out of the studied methods, temperature scaling produces the most well-calibrated image classification models.

Contents

1 Abbreviations

AECE	Adaptive Expected Calibration Error
AMCE	Adaptive Maximum Calibration Error
ECE	Expected Calibration Error
NLL	Negative Log-Likelihood
SGD	Stochastic Gradient Decent
SWA	Stochastic Weight Averaging
SWAG	Stochastic Weight Averaging Gaussian

2 Introduction

Deep neural networks have been shown to be successful in a wide range of supervised learning applications [1]. The ability to predict their certainty is however still unsatisfactory and limits the use in real-world applications. Together with the predicted class probabilities the network can output the confidence of the prediction [2], where an overconfident network produces too high class probabilities.

The predicted class probability of 0.80 for a well-calibrated network can be interpreted as the class appearing 80% of the time for that input. Hence, a well-calibrated network expresses its uncertainty through a low confidence or predicted probability.

The aim of this study is to evaluate methods for achieving well-calibrated image prediction models and a number of certainty measures. In the study both Bayesian and non-Bayesian methods are explored. The methods studied are temperature scaling [2], ensemble methods [3] and Stochastic Weight Averaging Gaussian [4]. The methods are applied to three different network structures of varying network depth. The studied certainty metrics are: Negative Log-Likelihood, Brier Score, Expected Calibration Error, Adaptive Expected Calibration Error and Adaptive Maximum Calibration Error.

3 Background

Supervised learning is the machine learning subbranch, where a model is trained using pairwise input and output signals. Neural networks are commonly trained using supervised learning and deeper network structures have in recent years achieved great success in a wide range of applications [1]. However, these deep neural network classifiers have been shown to lack the ability to estimate their own predictive certainty accurately. The deep networks tend to overestimate their certainty, in other words they provide overconfident models.

3.1 Bayesian Methods

Bayesian methods express the uncertainty of the model parameters $\boldsymbol{w}$ through probability distributions over the parameters [5]. The relationship between the posterior distribution $p(\boldsymbol{w}|\mathcal{D})$, the likelihood function $p(\mathcal{D}|\boldsymbol{w})$ and the prior distribution $p(\boldsymbol{w})$ can be expressed by Bayes' theorem as

$$\text{posterior} \propto \text{likelihood} \times \text{prior},$$

where the posterior distribution describes our knowledge of the parameters $\boldsymbol{w}$ after observing the data $\mathcal{D}$, the likelihood describes the probability of the data in view of the parameters and the prior distribution encodes prior belief of the parameters.

The natural certainty representation and ability to utilize prior knowledge are advantages of Bayesian methods [5]. But for modern neural networks Bayesian approaches have previously been intractable. Due to the high computational load associated with high number of parameters and non-convexity of the posterior [4]. Modern approaches to Bayesian deep learning include Variational Inference, Dropout Variational Inference, Laplace Approximation and Markov Chain Monte Carlo.

3.2 Calibration

A neural network can output both a class prediction and an indication of how confident the network is over its prediction, also known as its *confidence* [2]. For example if a network outputs 100 predictions, all with a confidence of 0.8. Then, if the network is well calibrated, the accuracy of the network should be 80% (i.e., around 80 of those 100 predictions should be correct).

Modern neural networks tend to be deeper, i.e. have more layers, to achieve a higher accuracy. However it has been shown that deeper neural networks also tend to be miscalibrated [2]. In order to achieve more calibrated networks one can use calibration methods, which are further discussed in Section 3.4.

A reliability diagram is a tool for visualizing the calibration of a model. An example of a reliability diagram is presented in Figure 1. A model's confidence given a sample is calculated as the maximum predicted class probability. For a perfectly calibrated model the confidence is equal to the accuracy. The samples are sorted and grouped (binned) according to the confidence value of each sample. The reliability diagrams are then constructed by plotting the average confidence against the accuracy in each bin. In this study a uniform and an adaptive binning strategy is implemented.

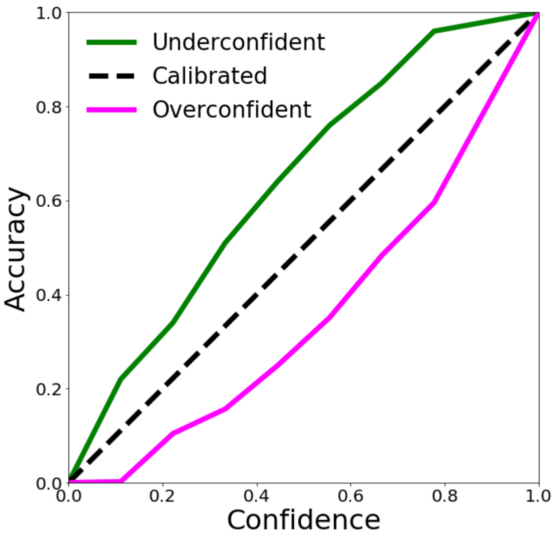


Figure 1: An example of a reliability diagram. The dashed line corresponds to a perfectly calibrated model. The green line corresponds to an underconfident model and the magenta line corresponds to an overconfident model.

3.3 Certainty Measures

Scoring rules measure the accuracy of probabilistic predictions by assigning a score based on the predicted outcome and the true outcome [6]. Let S be a scoring rule, then the score is given by $S(P, Q)$, where P is the predicted outcome and Q is the true outcome. The score is presented as a value within the interval [0, 1].

A scoring rule is proper if the expected score is optimized by the true probability distribution [6]. Depending on the type of scoring rule, which can be transformations of each other, the goal can be to either maximize or minimize the expected score. As an example the goal for the Brier Score is to minimize the expected score.

3.3.1 Negative Log-Likelihood

The Negative Log-Likelihood (NLL) is a proper scoring rule [7]. It is defined as

$$-\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk},$$

where N is the number of samples, K the number of classes, t_{nk} is element k from a one-hot encoded target vector which corresponds to sample n and y_{nk} is the predicted probability for sample n and class k [5].

In multiclass classification this measure is also known as the cross-entropy error function [2].

3.3.2 Brier Score

The Brier Score is a commonly used proper scoring rule which can be used as a calibration measure [7]. The Brier Score is defined as:

$$\frac{1}{N}\sum_{n=1}^{N}(y_n - t_n)^2,$$

where N is the number of samples, y_n is the predicted probability and t_n is the label for sample n [8].

3.3.3 Binning Based Measures

Expected Calibration Error (ECE) [9], is a certainty measure defined as

$$\mathop{\mathbb{E}}_{\hat{P}}\left[\text{abs}\left(\mathbb{P}\left(\hat{Y} = Y|\hat{P} = p\right) - p\right)\right],$$

where $\hat{Y}$ is a class prediction, $\hat{P}$ its associated confidence, Y the label and p is the confidence [2]. It can be described as the difference in the expected confidence and accuracy. A low value of ECE indicates a more well-calibrated model.

Adaptive Expected Calibration Error (AECE) is a certainty measure which extends ECE. Unlike ECE, which uses either equally sized bins or bins with equal range, AECE uses an adaptive method to determine the size of the bins [10]. The bin size is smaller in areas of higher population. It is claimed in [10] that AECE is better than ECE at capturing the calibration error due to its adaptive binning. Adaptive Maximum Calibration Error (AMCE) is similar to AECE but uses an adaptive binning to find the maximum expected calibration error in any bin [10].

3.4 Methods for Achieving Well-Calibrated Models

This section introduces the three studied methods for achieving well-calibrated models: temperature scaling, ensemble methods and Stochastic Weight Averaging Gaussian.

3.4.1 Temperature Scaling

Temperature scaling is a calibration method which improves the certainty measures of neural networks [11]. The method is based on Platt Scaling [12] and is presented in [2]. One scalar parameter T is used for rescaling the non-probabilistic input to the softmax layer. The aim of the non-Bayesian method is to make the predictions of a neural network more calibrated. The calibrated probability can be expressed as follows:

$$\hat{q}_i = \max_k \sigma_{SM} \left(\frac{\mathbf{z}_i}{T} \right)^{(k)},$$

where $\mathbf{z}_i$ is the logit vector for the sample i, σ_{SM} is the softmax function, T is the scalar parameter often referred to as the temperature and k is the class index. The parameter T has the same value across all classes with the condition that $T > 0$ and is trained using the loss function NLL. Temperature scaling does not affect the accuracy of the model since including the scalar parameter does not change the predictions of the network.

The results from [2] show that “modern” neural networks tend to be more miscalibrated. Although the accuracy is usually increased by using a more “modern” neural networks, they are also overconfident due to being miscalibrated. This implies that the network’s confidence is greater than its actual accuracy. In [2] multiple post-processing calibration methods are evaluated and compared amongst each other as well as to temperature scaling. It is the parametric approach of temperature scaling which differs it from many other calibration methods.

The certainty measures used in the evaluation are ECE, maximum calibration error and NLL [2]. The calibration methods for multi-class models evaluated are the following: extensions of several binning methods, matrix, vector and temperature scaling. Matrix, vector and temperature scaling are all extensions of Platt scaling where temperature scaling is the simplest approach. Using temperature scaling as the calibration method resulted in significantly more well-calibrated networks for some datasets compared to the other methods. For the rest of the studied datasets temperature scaling had similar performance as the other methods.

3.4.2 Ensemble Methods

Ensemble methods is an approach where a combination of multiple models are used to obtain a stronger model and hence achieve a better predictive performance [3]. In [7] ensemble methods and two measures are proposed for evaluation: calibration and out-of-distribution examples, where out-of-distribution examples refer to if the ”network knows what it knows”.

The method is divided into three separated parts:

1. *Proper scoring rules* are used as the benchmark for training
2. *Adversarial training* to smoothing the distributions
3. *Ensemble* training

and are executed in this order [7].

Proper scoring rules are used for measuring the property of calibration. To measure calibration the Brier Score and NLL are used [7]. To obtain a smoother predictive distribution, virtual adversarial training is mentioned as a complementary alternative proposed in [13]. A randomization-based approach is used for training the ensemble, which is the last step of the proposed process. An example of a randomization-based method is random forest, which can be trained in parallel.

When performing the experiment, the proper scoring rule NLL is evaluated [7]. Both the Brier Score and accuracy are used as measures for the classification problems, while the root mean square error is used as a measure for the problems involving regression.

3.4.3 Stochastic Weight Averaging Gaussian - SWAG

Stochastic Weight Averaging Gaussian (SWAG) is an approximate Bayesian inference technique for deep learning [4]. SWAG is an extension of Stochastic Weight Averaging (SWA) [14], where weights are averaged over Stochastic Gradient Descent (SGD) iterates. A high and constant learning rate schedule is used to improve generalization.

SWAG approximates the posterior distribution of the neural network weights by a Gaussian [4]. The first moment is taken from the SWA solution and the covariance matrix is calculated as a low rank matrix plus a diagonal matrix. Bayesian model averaging is then performed by sampling from the approximate distribution.

The covariance matrix approximation used in SWAG is more flexible than the standard diagonal matrices used in variational inference and Laplace approximations in Bayesian deep learning [4]. As previously mentioned, the

SWAG approximation is a sum of a diagonal and a low rank matrix $\frac{1}{2} \cdot (\Sigma_{\text{diag}} + \Sigma_{\text{low-rank}})$. The rank of the approximated matrix is restricted by the hyperparameter K, which corresponds to the last K number of epochs of training considered. The final approximated posterior distribution can be written as $\mathcal{N}(\theta_{\text{SWA}}, \frac{1}{2} \cdot (\Sigma_{\text{diag}} + \Sigma_{\text{low-rank}}))$, where θ_{SWA} is the SWA solution.

If SGD with weight decay or explicit L2 regularization is used for training deep neural networks, then SWAG can be applied without modifications to training [4]. For training a piecewise constant learning rate schedule and an adaptive moment is used. To evaluate uncertainty the measures NLL and ECE are used, calibration is investigated by studying reliability diagrams. Experiments on different computer vision tasks are conducted and the performance compared to alternatives like MC dropout, temperature scaling and KFAC Laplace. The results indicate that SWAG performs well compared to the other methods.

4 Implementation

First a baseline model with a test accuracy of approximately 90% was constructed for each network. The chosen networks were LeNet-5 [15], VGG-16 [16] and ResNet-50 [17], where the number indicates the number of layers in the network. The open source machine learning platform TensorFlow [18] and the neural networks API Keras [19] were used for the implementation. The code was written in Python and is available at the public GitHub repository [20]. The computations were made on a NVIDIA-SMI 440.44 GPU with CUDA 10.1, via Google Colab [21]. The hyperparameters used for training the baseline are presented in Table 1. SGD was used as optimizer where the learning rate was set to 0.01 with a decay of 0.0001 and the momentum was set to 0.9. Cross-entropy was used as the loss function.

4.1 Networks & Dataset

For each network the training size was chosen to achieve the desired accuracy of approximately 90%. The MNIST dataset, consisting of handwritten digits, was used for training and testing our models [22]. The dataset contains 70 000 images of size 28x28 pixels. The test size was kept constant in order to perform a fair comparison. The number of epochs as well as the batch sizes for training and testing were also kept constant. The final values for these hyperparameters are presented in Table 1.

Table 1: Setup for training and testing the implemented networks with the MNIST dataset. The number of epochs differs when using SWAG. These settings were chosen to achieve an accuracy of approximately 90%.

	Training size	Test size	Epochs	Batch Size (training/test)
LeNet-5	900	40 000	10	64/32
VGG-16	5 000	40 000	10	64/32
Resnet-50	6 000	40 000	10	64/32

4.2 Methods for Achieving Well-Calibrated Models

The implementation of temperature scaling was based on code available at [23]. The implementation required the logits of the network. These were calculated using a function that inverted the softmax operation and hence returned the logits [24]. The temperature T, was then found by minimizing the cross-entropy of the logits scaled with the parameter T and the correct labels. The same dataset used for training the baseline model was used when determining the parameter T. The chosen optimizer for this step implemented the momentum algorithm. The value of the momentum was set to 0.9 and the learning rate to 0.01

Ten ensemble members were used in the ensemble methods implementation, and each ensemble member were one of the three network structures previously mentioned. This is in contrast to the approach in [1] and [3] where multilayer perceptrons with 2-3 hidden layers were used. Adversarial training was not implemented in this study.

The ensemble members were kept if the accuracy was equal to or greater than 0.2, to make sure that the ensemble member performs at least better than choosing a label at random (1/10 for MNIST). The predictions of the ensemble was obtained by taking the mean of the predictions of the separate ensemble members. As a consequence weights for all ensemble members must be kept in order to make new predictions. Hence, the memory usage is directly proportional to the number of ensemble members.

The implementation of SWAG was based on code available at [25]. The networks LeNet-5 and VGG-16 were trained with SWAG. The original implementation in [4] uses an optimizer with a varying momentum as well as a learning rate schedule. The varying learning rate schedule was implemented, but a fixed momentum of 0.9 was used. When these features were added to the original implementation [25] the learning process failed as the optimizer got stuck in a local optima. To solve this issue, while still keeping the learning rate schedule, the momentum was reduced to 0.5 for VGG-16.

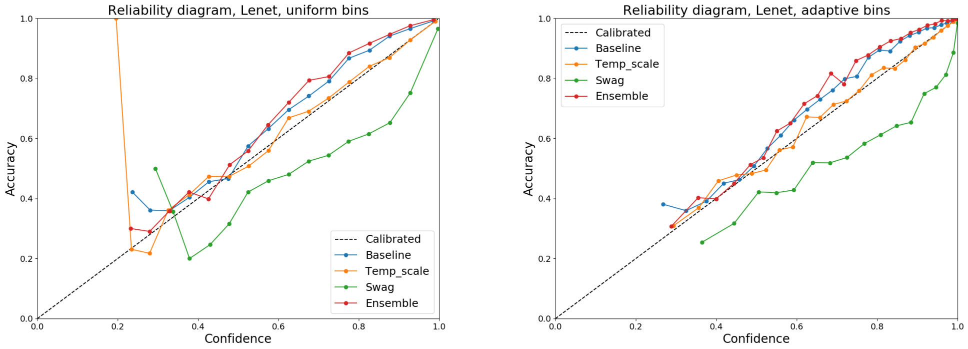
The dataset in this study is significantly smaller than the original paper. To account for this the number of training epochs were reduced. In the original paper the number of epochs is set to 300, the number of SWAG epochs to 160 and the hyperparameter K set to 20. As not to interfere with the method itself and keep K constant the number of epochs was set to 120 and the number of SWAG epochs to 61.

5 Results

This section presents the results of applying temperature scaling, ensemble methods and SWAG to three different neural network structures. The results are presented in the form of tables and reliability diagrams. The five implemented metrics are presented in Tables 2-4 and the best model according to each metric is highlighted. For each network reliability diagrams based on both uniform and adaptive binning are presented. The reliability diagrams include a line representing a perfectly calibrated model. Both the tables and the figures include a baseline model for each network as a reference. Enlarged reliability diagrams are located in Appendix A.

5.1 LeNet-5

The reliability diagrams in Figure 2 suggest that SWAG produces an overconfident model for LeNet-5. Both the baseline and the ensemble methods produces underconfident models while temperature scaling results in the most well-calibrated model.



(a) Uniform binning.

(b) Adaptive binning.

Figure 2: Reliability diagrams for LeNet-5 with uniform and adaptive binning respectively. The results for all three studied methods can be seen as well as the baseline. The result for a perfectly calibrated model is plotted for reference.

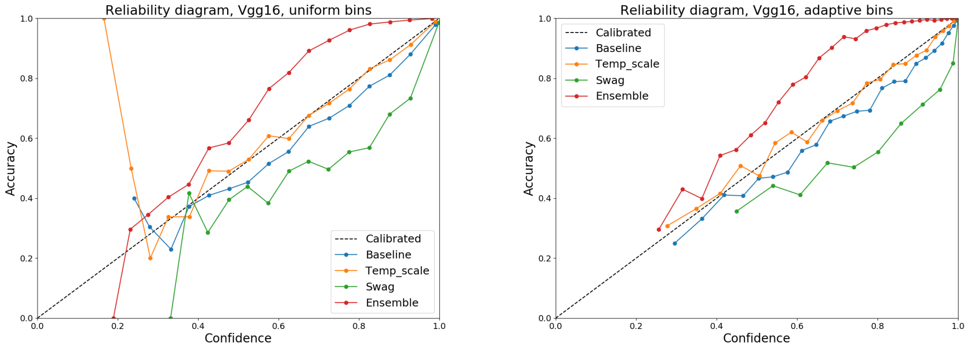
It can be seen in Table 2 that neither NLL nor the Brier Score captures a difference in calibration with temperature scaling compared to the baseline for LeNet-5. These metrics suggest that ensemble methods improve the calibration while the binning based metrics suggest an impaired calibration. Brier does not agree that the calibration is impaired by using SWAG.

Table 2: Results for LeNet. Used training size = 900, test size = 40 000, learning rate = 0.01, learning rate decay = 0.0001. For SWAG a decreasing learning rate schedule was used. The best method according to each metric is highlighted.

	Baseline	**Temp. Scaling**	**Ensemble**	**SWAG**
NLL	0.30	0.30	0.25	0.38
Brier	0.14	0.14	0.12	0.13
ECE	0.030	0.0050	0.039	0.049
AECE	0.027	0.0052	0.039	0.050
AMCE	0.11	0.053	0.12	0.23

5.2 VGG-16

The reliability diagrams in Figure 3 suggest that both SWAG and the baseline produce overconfident models for VGG-16. Ensemble methods result in an underconfident model while temperature scaling produces the most well-calibrated model.



(a) Uniform binning.

(b) Adaptive binning.

Figure 3: Reliability diagrams for VGG-16 with uniform and adaptive binning respectively. The results for all three studied methods can be seen as well as the baseline. The result for a perfectly calibrated model is plotted for reference.

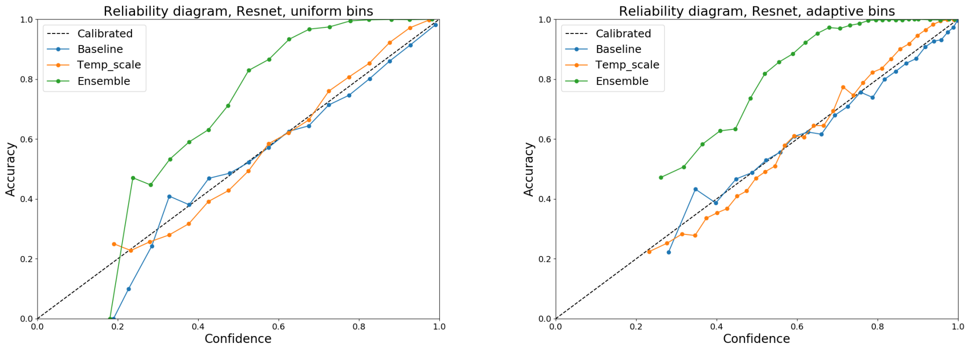
It can be seen in Table 3 that for VGG-16 NLL and Brier agree that all methods improve the calibration. The binning based metrics suggest that the ensemble methods impair the calibration. All metrics agree that temperature scaling improves the calibration for VGG-16.

Table 3: Results for VGG-16. Used training size = 5 000, test size = 40 000, learning rate = 0.01, learning rate decay = 0.0001. For SWAG a decreasing learning rate schedule was used. The best method according to each metric is highlighted.

	Baseline	**Temp. Scaling**	**Ensemble**	**SWAG**
NLL	0.35	0.22	0.22	0.16
Brier	0.16	0.10	0.093	0.050
ECE	0.025	0.0051	0.079	0.020
AECE	0.025	0.0064	0.079	0.020
AMCE	0.090	0.050	0.22	0.25

5.3 ResNet-50

The reliability diagrams in Figure 4 suggests that ensemble methods produces an underconfident model for ResNet-50. Both the baseline and temperature scaling produce the most well-calibrated models.



(a) Uniform binning.

(b) Adaptive binning.

Figure 4: Reliability diagrams for ResNet-50 with uniform and adaptive binning respectively. The results for the applied methods can be seen as well as the baseline. The result for a perfectly calibrated model is plotted for reference.

For ResNet-50 it can be seen in Table 4 that ensemble methods is the preferred method according to NLL and Brier. According to ECE and AECE the baseline is to be preferred while AMCE suggests that temperature scaling is the preferred method.

Table 4: Results for ResNet-50. Used training size = 6 000, test size = 40 000, learning rate = 0.01, learning rate decay = 0.0001. The best method according to each metric is highlighted.

	Baseline	**Temp. Scaling**	**Ensemble**
NLL	0.31	0.81	0.25
Brier	0.15	0.38	0.091
ECE	0.012	0.031	0.15
AECE	0.012	0.032	0.15
AMCE	0.085	0.068	0.30

5.4 Accuracy, Loss & Computational Time

The accuracy and loss for the different combinations of implemented networks and methods are presented in Table 5. The baseline is kept for reference. SWAG was not implemented for ResNet-50 hence there is no result for this combination in the table.

Table 5: Accuracy and loss for combinations of implemented networks and methods. The first value is the accuracy and the second value is the loss.

	Baseline	**Temp. Scaling**	**Ensemble**	**SWAG**
LeNet-5	0.91/0.30	0.91/0.31	0.91/0.33	0.92/0.38
VGG-16	0.89/0.35	0.93/0.22	0.92/0.27	0.97/0.16
ResNet-50	0.90/0.31	0.73/0.81	0.96/0.15	-

Temperature scaling should be performed on the pretrained baseline and hence not affect the accuracy of the network. However this was not how the implementation was done, which can be seen in Table 5 as the accuracy of the baseline model and the scaled model are not equal.

In Table 6 the run times are presented for LeNet-5, VGG-16 and ResNet-50. The values correspond to the training time and evaluation time respectively. The ensemble members were trained sequentially, not in parallel which would have greatly reduced the run time. The values presented in Table 6 are therefore presented per ensemble member instead of the total run time. The run times for applying SWAG to the studied networks are significantly greater than for the other methods.

Table 6: Training and evaluation time for combinations of implemented networks and methods. The first value is the training time and the second value is the evaluation time. For the ensemble the presented time is an average per ensemble member.

	Baseline [s]	**Temp. Scaling [s]**	**Ensemble [s]**	**SWAG [min]**
LeNet-5	8/2	3/3	2/3	1.8/1.4
VGG-16	40/19	39/18	39/19	6.4/4.1
ResNet-50	80/19	88/23	111/39	-

5.5 Measures & Binning Strategies

The use of uniform or adaptive binning does not seem to have a great impact, as the two certainty measures ECE and AECE produce similar values. The point corresponding to the first bin in the reliability diagrams with uniform binning deviates for all three networks. This is likely caused by a low number of data points in the first bin. This is not present in the adaptive diagrams, and showcases an advantage of the adaptive binning strategy.

The binning based measures are consistent with the reliability diagrams except for ResNet-50, where AMCE deviates. The inconsistency can be explained by the definition of AMCE, which picks the maximum calibration error in any bin with adaptive binning. In Figure 4 it can be seen that the baseline has an outlier value which causes the deviating choice of method. Meanwhile, ECE and AECE captures the average deviation from the perfectly calibrated model, which is smaller for the baseline.

The non-binning based measures consistently agree regarding the most preferred methods. But the preferred model is constantly in a disagreement with the conclusions drawn form the reliability diagrams. These metrics either suggest ensemble methods or SWAG, which is never proposed by the diagrams nor binning based measures.

6 Discussion & Conclusions

A challenge within this field is the lack of consensus regarding the choice of certainty metrics and baseline models. Previous studies claim that deeper neural networks are less well-calibrated. However this is inconsistent with the results presented in this study. The baselines deviate less from the perfectly calibrated line in the reliability diagrams and the certainty measures tend to decrease for the deeper networks. Less well-calibrated baseline models might better showcase the performance of the calibration methods.

In Section 5.5 it was suggested that the binning based metrics give more consistent results for different networks. However the basis of this conclusion can be considered biased as the reliability diagrams and the binning based metrics are based on the same binning strategy. Hence, it is not surprising that they agree. In a way one could consider them two different ways of viewing the same information, where ECE corresponds to the uniform binning reliability diagrams and AECE to the adaptive binning reliability diagrams. An unexpected conclusion of this study is that the deep baseline models, without any form of calibration, were fairly well-calibrated. This conclusion was also based on the reliability diagrams and the binning based measures and might therefore also be biased.

Temperature scaling was the only method that improved the calibration of the networks, according to the reliability diagrams. For ResNet-50 all the measures except AMCE indicate a reduced performance when applying temperature scaling. Advantages of temperature scaling include that it is fast to train and straightforward to implement.

From the reliability diagrams and the binning based measures it can be concluded that the ensemble methods result in underconfident predictions. A possible explanation is that our baseline models are relatively well-calibrated. The calibrated predictive probabilities are calculated as an average over several networks. Since the accuracies of the networks varies, the class probabilities (which sum to 1) are distributed differently over the classes. Hence, it is not surprising that applying ensemble methods to a well-calibrated model produces underconfident predictions. Had the original network structures been overconfident then the method might have worked as expected. The non-binning based certainty measures suggest that the ensemble methods improve the calibration of the networks. The original paper suggests ensembles of multilayer perceptrons, as opposed to deep convolutional networks. This approach was not evaluated and might still produce well-calibrated predictions.

SWAG produces overconfident models for both LeNet-5 and VGG-16 according to the reliability diagrams and the binning based measures. The non-binning based measures gave inconsistent results and as ResNet-50 was not implemented it is difficult to assess the method. Implementing SWAG was challenging, the training process tended to get stuck in local optima and the training time was by far the longest. Hence we do not recommend this method. However, due to the differences in implementation compared to the original paper mentioned in Section 4.2, the evaluation of the method might be considered unfair.

In conclusion the binning based metrics give more consistent results. This combined with the generated reliability diagrams lead to the conclusion that temperature scaling is the best studied method for achieving well-calibrated image classification models.

7 Future Work

There are several interesting topics to further explore within this field. One would be to study deeper network structures, to further investigate how the performance of the methods varies. Another topic is to explore other datasets and less well-calibrated baseline models.

8 Acknowledgements

We would like to take the opportunity to thank our supervisors Joakim Lindblad and Nataša Sladoje at the Department of Information Technology, Division of Visual Information and Interaction, Uppsala University.

References

[1] Yaniv Ovadia et al. "Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift". In: (2019). arXiv: 1906.02530.

[2] Chuan Guo et al. "On Calibration of Modern Neural Networks". In: *CoRR* abs/1706.04599 (2017). arXiv: 1706.04599. URL: http://arxiv.org/abs/1706.04599.

[3] Terence J. OâKane and Jorgen S. Frederiksen. "A Comparison of Statistical Dynamical and Ensemble Prediction Methods during Blocking". In: *Journal of the Atmospheric Sciences* 65.2 (2008), pp. 426–447. DOI: 10.1175/2007JAS2300.1. URL: https://doi.org/10.1175/2007JAS2300.1.

[4] Wesley Maddox et al. "A Simple Baseline for Bayesian Uncertainty in Deep Learning". In: *arXiv preprint arXiv:1902.02476* (2019).

[5] Christopher M Bishop. *Pattern recognition and machine learning.* Springer Science + Business Media, 2006.

[6] Tilmann Gneiting and Adrian Raftery. "Strictly Proper Scoring Rules, Prediction, and Estimation". In: *Journal of the American Statistical Association* 102 (Mar. 2007), pp. 359–378.

[7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". In: *Advances in Neural Information Processing Systems 30.* Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 6402–6413. URL: http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf.

[8] Glenn W. Brier. "Verification of Forecasts Expressed in Terms of Probability". In: *Monthly Weather Review* 78.1 (1950), pp. 1–3. URL: https://doi.org/10.1175/1520-0493(1950)078%3C0001:VOFEIT%3E2.0.CO;2.

[9] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. "Obtaining well calibrated probabilities using bayesian binning". In: *Twenty-Ninth AAAI Conference on Artificial Intelligence.* 2015.

[10] Yukun Ding et al. "Evaluation of Neural Network Uncertainty Estimation with Application to Resource-Constrained Platforms". In: (2019). arXiv: 1903.02050.

[11] Azadeh Mozafari et al. "A New Loss Function for Temperature Scaling to have Better Calibrated Deep Networks". In: (Oct. 2018).

[12] John C. Platt. “Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods”. In: (1999), pp. 61–74.

[13] Takeru Miyato et al. *Distributional Smoothing with Virtual Adversarial Training*. 2015. arXiv: 1507.00677.

[14] Pavel Izmailov et al. “Averaging weights leads to wider optima and better generalization”. In: *arXiv preprint arXiv:1803.05407* (2018).

[15] Yann LeCun et al. “Gradient-based learning applied to document recognition”. In: 1998.

[16] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: 1409.1556.

[17] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385.

[18] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/. Software available from tensorflow.org. 2015.

[19] François Chollet et al. *Keras*. https://keras.io. 2015.

[20] Melanie Andersson, Sara Hedar, and Andreas Isaac. *Estimating Certainty of Deep Learning*. https://github.com/Andreas12321/Est-Cert-Final. 2020.

[21] *Google Colaboratory*. https://colab.research.google.com. Accessed: 2020-01-06.

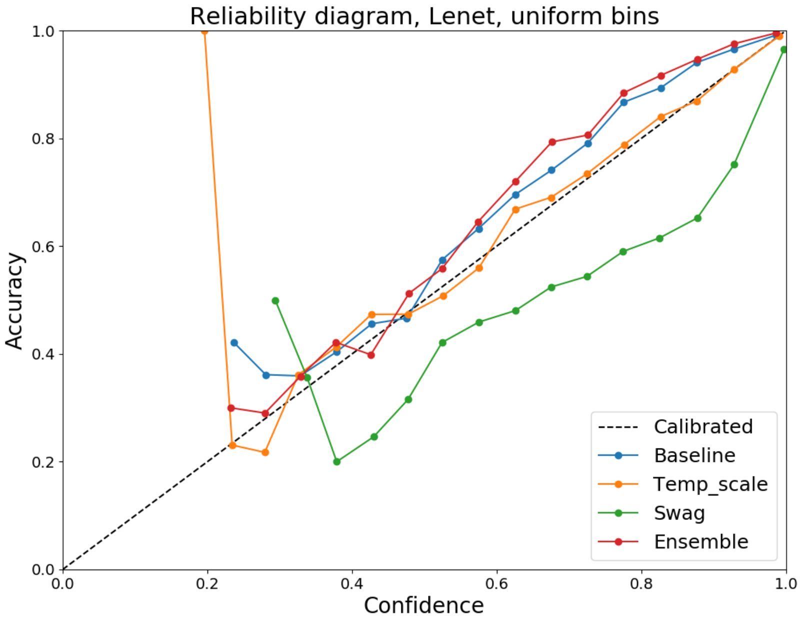
[22] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: http://yann.lecun.com/exdb/mnist/.

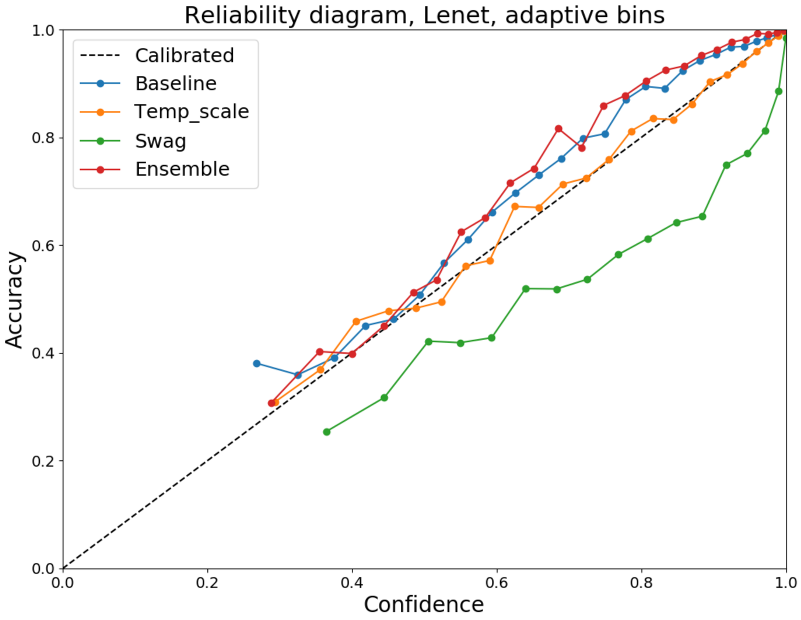
[23] Ondrej Biza. *Confidence calibration for neural networks*. https://github.com/ondrejba/tf_calibrate. 2018.

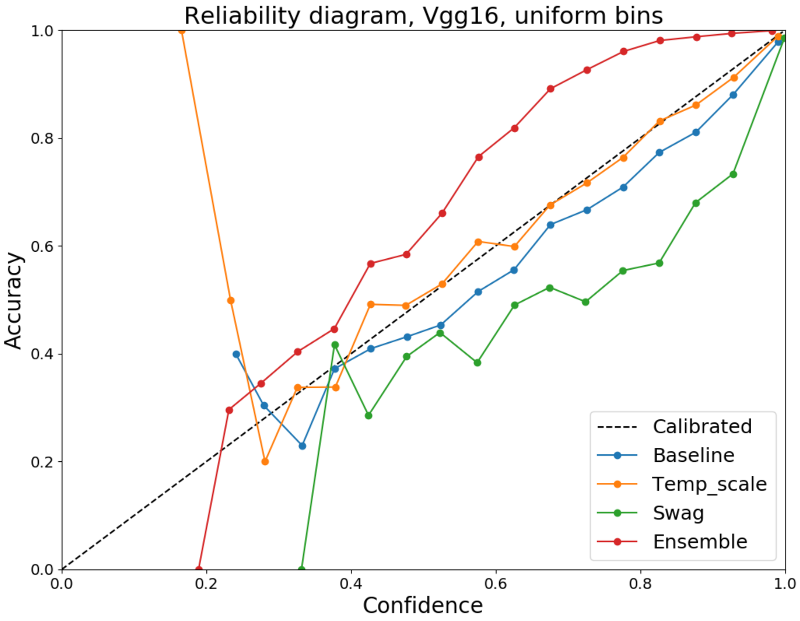
[24] Google AI Research. *Can You Trust Your Model’s Uncertainty?* https://github.com/google-research/google-research/tree/master/uq_benchmark_2019. 2019.

[25] Josef Malmström, Victor Löfgren, and Najib Yavari. *swag-reproduced*. https://github.com/DD2412-Final-Projects/swag-reproduced/tree/authors-config. 2019.

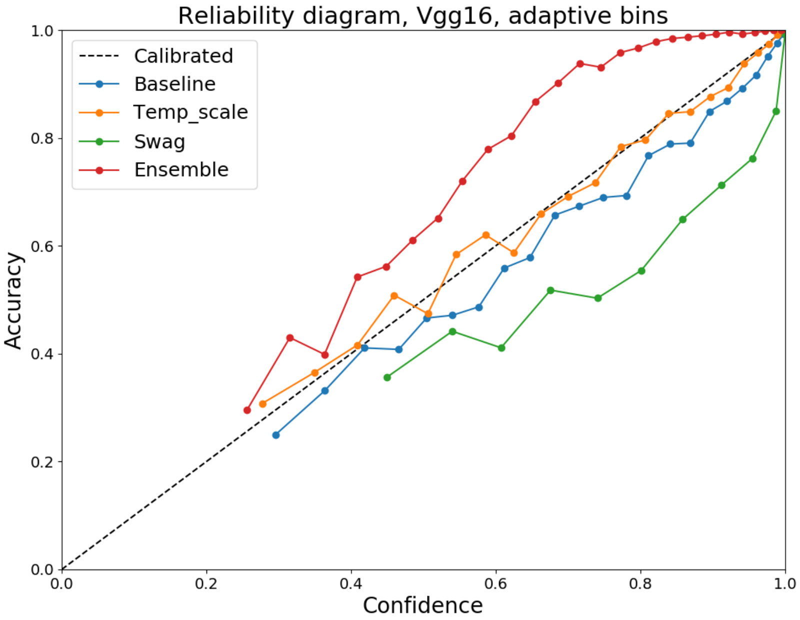
A Appendix


Reliability diagram, Lenet, uniform bins
Accuracy
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
Calibrated
Baseline
Temp_scale
Swag
Ensemble

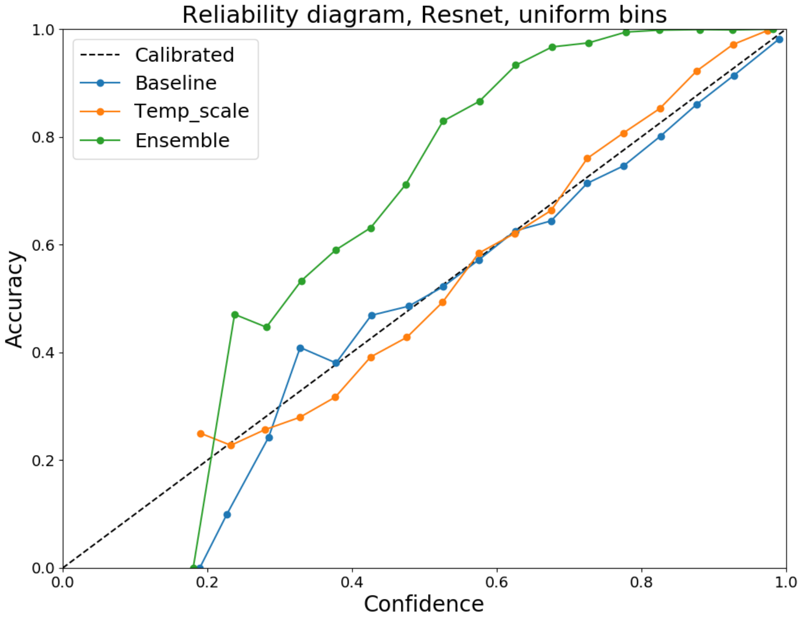

Reliability diagram, Lenet, adaptive bins
Accuracy
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
Calibrated
Baseline
Temp_scale
Swag
Ensemble

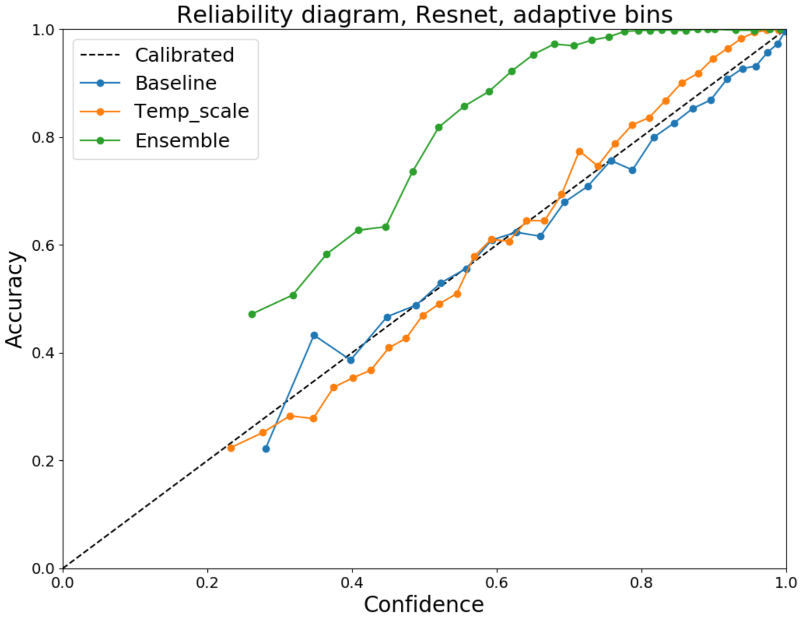


Reliability diagram, Vgg16, uniform bins
Accuracy
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Calibrated
Baseline
Temp_scale
Swag
Ensemble



Reliability diagram, Vgg16, adaptive bins
Accuracy
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Calibrated
Baseline
Temp_scale
Swag
Ensemble


Reliability diagram, Resnet, uniform bins
Calibrated
Baseline
Temp_scale
Ensemble
Accuracy
Confidence
0.0
0.2
0.4
0.6
0.8
1.0


Reliability diagram, Resnet, adaptive bins
Calibrated
Baseline
Temp_scale
Ensemble
Accuracy
Confidence
0.0
0.2
0.4
0.6
0.8
1.0