Journal of Physics:
Conference Series

PURPOSE-LED PUBLISHING™

PAPER • **OPEN ACCESS**

Monte Carlo averaging for uncertainty estimation in neural networks

To cite this article: Cedrique Rovile Njieutcheu Tassi *et al* 2023 *J. Phys.: Conf. Ser.* **2506** 012004

View the article online for updates and enhancements.

You may also like

- Localized sp–d exchange interaction in ferromagnetic $Ga_{1-x}Mn_xAs$ observed by magnetic circular dichroism spectroscopy of *L* critical points
H Tanaka, W M Jadwisienczak, H Saito et al.
- The progress and perspectives of terahertz technology for diagnosis of neoplasms: a review
K I Zaytsev, I N Dolganova, N V Chernomyrdin et al.
- Quantitative measurement of magnetic parameters by electron magnetic chiral dichroism
Dong-Sheng Song, , Zi-Qiang Wang et al.

This content was downloaded from IP address 80.180.30.211 on 12/02/2025 at 15:51

Monte Carlo averaging for uncertainty estimation in neural networks

Cedrique Rovile Njieutcheu Tassi[1*], Anko Börner[1], and Rudolph Triebel[2]

[1] German Aerospace Center, Institute of Optical Sensor Systems, Berlin, Germany
[2] German Aerospace Center, Institute of Robotics and Mechatronics, Wessling, Germany
[*] E-mail: njieutcheu@gmail.com

Abstract. Although convolutional neural networks (CNNs) are widely used in modern classifiers, they are affected by overfitting and lack robustness leading to overconfident false predictions (FPs). By preventing FPs, certain consequences (such as accidents and financial losses) can be avoided and the use of CNNs in safety- and/or mission-critical applications would be effective. In this work, we aim to improve the separability of true predictions (TPs) and FPs by enforcing the confidence determining uncertainty to be high for TPs and low for FPs. To achieve this, we must devise a suitable method. We proposed the use of Monte Carlo averaging (MCA) and thus compare it with related methods, such as baseline (single CNN), Monte Carlo dropout (MCD), ensemble, and mixture of Monte Carlo dropout (MMCD). This comparison is performed using the results of experiments conducted on four datasets with three different architectures. The results show that MCA performs as well as or even better than MMCD, which in turn performs better than baseline, ensemble, and MCD. Consequently, MCA could be used instead of MMCD for uncertainty estimation, especially because it does not require a predefined distribution and it is less expensive than MMCD.

Keywords: Convolutional neural network (CNN), ensemble, Monte Carlo dropout (MCD), mixture of Monte Carlo dropout (MMCD), Monte Carlo averaging (MCA), separating true predictions (TPs) and false predictions (FPs), confidence calibration

1. Introduction

Because of the emergence of large datasets, increasing computational power, and advances in deep learning, convolutional neural networks (CNNs) have become the standard for solving classification problems. Despite the widespread use of CNNs in modern classifiers [1, 2, 3], they are faced with several problems, such as overfitting [4], which causes overconfident predictions [5], and lack of robustness. An example of a lack of robustness was described by Hendrycks and Dietterich [6], who empirically showed that CNNs can change their predictions when perturbations, such as blur or noise, are applied to the input image. This overconfidence and lack of robustness can lead to overconfident FPs. Several other authors also empirically showed that CNNs can overconfidently misclassify out-of-domain (OOD) examples (situations not present in the training data) [7, 5, 8]. In [9], the authors showed that CNNs can also overconfidently misclassify *domain-shift examples*, which are in-domain examples (situations present in the training data) affected by a set of perturbations, such as changes in the camera lens and lighting conditions. Overconfident FPs can be costly and dangerous, especially when


Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

CNN-based classifiers are part of the decision making unit of systems for safety- and/or mission-critical applications, such as collision avoidance [10], door recognition for visual-based robot navigation [11], and pedestrian detection [12]. In this study, FPs will result in false actions in the environment, leading to robot collisions, potential false medical treatments, and/or increased financial costs. By preventing FPs, we can avoid these consequences and encourage the widespread adoption of CNNs in safety- and/or mission-critical applications. This goal will be achieved by estimating and evaluating the predictive uncertainty of CNNs and ensuring that the confidence measuring uncertainty is high ($[50\%, 100\%]$) for TPs and low ($[0\%, 50\%[$) for FPs. This, however, calls for a research question: *What method is required to achieve high and low confidence of CNN-based classifiers for TPs and FPs, respectively?* In our paper titled A Survey of Uncertainty in Deep Neural Networks [13], we suggested an uncertainty estimation technique that combines the strengths of both Bayesian and ensemble principles. This technique was adopted in this current work and thus, MCA was proposed, a method similar to the mixture of Monte Carlo dropout (MMCD), which combines the strengths of an ensemble and Monte Carlo dropout (MCD). MCA is deterministic similar to an ensemble, evaluates multiple features such as in the ensemble and MMCD, and evaluates uncertainty associated with extracted features similar to MCD and MMCD, both of which are stochastic. We, therefore, empirically compared MCA and other related methods (baseline (single CNN), MCD, ensemble, and MMCD) based on results from experiments conducted on four datasets (CIFAR10, FashionMNIST, MNIST, and GTSRB) using three different architectures (DenseNets, ResNets, and VGGNets). Results show that MCA can perform equally or even better than MMCD. Similar to MMCD, MCA can preserve the classification accuracy of the underlying ensemble, which can increase the classification accuracy of the baseline, which can only be preserved via MCD. Similar to MMCD, MCA can separate TPs and FPs better than baseline, ensemble, and MCD but at the cost of increasing calibration error on test data.

2. Related works

MCD [14, 15] is one of the most widely used approximations for Bayesian inference. It samples features by dropping neurons using Bernoulli masks. Several related works have investigated various extensions of MCD. For example, Tassi [16] investigated the use of dropout in pooling and/or convolution instead of fully-connected layers. Zeng et al. [17] investigated the position and number of Bayesian layers required to approximate a fully Bayesian neural network (BNN) and found that only a few Bayesian layers near the output of the BNN are sufficient. Similarly, Brosse et al. [18] evaluated the quality of uncertainty that results when only the last layer is Bayesian, and found that last-layer BNNs perform similarly well compared with full BNNs. Kristiadi et al. [19] complemented the empirical evidence of Brosse et al. [18] with a theoretical justification showing why it is sufficient to make the last layer Bayesian at low cost overhead. According to Zeng et al. [17] and further supported by Brosse et al. [18], the use of multiple Bayesian layers in a BNN can compromise accuracy without improving the quality of uncertainty. However, the more Bayesian layers are, for example, by using a high dropout probability, the more uncertainty we can capture at the cost of sacrificing the accuracy [17, 16]. Other studies investigated the use of different dropout strategies, such as drop-connect, where connections are dropped instead of neurons [20, 21], or structured dropout, where layers or channels are dropped [22]. Other studies [21, 16] evaluated the use of different dropout masks, such as Gaussian, Bernoulli, or a cascade of Gaussian and Bernoulli. Taken together, all these works show that a single MCD layer near the output of a CNN, for example, at the input of the first fully-connected layer, is sufficient for uncertainty quantification. Moreover, all the studies show that the MCD is sensitive to the sampling masks drawn from a predefined distribution. The proposed MCA does not require a predefined distribution from which masks are drawn and therefore overcomes the drawback of the MCD.

The ensemble was initially proposed for improving accuracy [23, 1]. Existing methods for introducing diversity among ensemble members, such as random initialization, data shuffling, bagging, and data augmentation, were originally proposed for improving the accuracy. Nevertheless, ensembles have become a popular method for uncertainty estimation through the pioneering work of Lakshminarayanan et al. [5]. Several related works have evaluated the performance of ensembles in capturing in-domain uncertainties [24] or OOD uncertainties [9]. Other works [25, 26, 9] compared ensembles with other uncertainty estimation methods such as MCD and concluded that ensembles perform better than MCD. Lakshminarayanan et al. [5] used random initialization and data shuffling to build ensembles for uncertainty estimation. Lee et al. [27] experimentally showed that the diversity introduced into ensemble members via random parameter initialization is more useful than that introduced via bagging. They concluded that random initialization is not only sufficient, but also preferable to bagging for building ensembles of CNNs because CNNs have a large parameter space and require large training data. They also showed that bagging can result in poorly calibrated ensembles. According to Wen et al. [28], data augmentation approaches, such as mixup [29], can also harm the calibration of ensembles. This has also been reported in other studies [30, 31]. In [31, 24, 32], the authors improved the calibration of ensembles using temperature scaling. In this work, to avoid harming the calibration of ensembles, diversity was introduced into ensemble members using random initialization, data shuffling, and standard label-preserving data augmentation techniques, such as rotation, translation, flipping, shear and additive Gaussian noise.

MMCD was used in [10, 33, 34] for uncertainty estimation. It combines the strengths of MCD and ensemble. Although MCD evaluates a single local optimum in a given solution space, but additionally considers the uncertainty of the local optimum, an ensemble includes multiple deterministic CNNs representing different local optima in the solution space and therefore evaluates multiple modes (extracted features) [35]. However, an ensemble does not account for the uncertainty around the individual modes. To explore the uncertainty around each mode, MMCD applies MCD to each ensemble member.

3. Background

3.1. Convolutional neural network

For image classification, a CNN is a function f that maps an input image $x \in \mathbb{R}^{H\times W\times C}$ to a class label $y \in U^K$, where H, W, and C are the height, weight, and number of channels of the input image, respectively. U^K and K denote the set of standard unit vectors of $\mathbb{R}^K$ and the number of possible classes, respectively. A CNN consists of two main modules: a *features extractor*, which is realized using convolutional and pooling layers, and a *discriminator*, which is realized using fully-connected layers. Thus, a CNN is a composite of two functions $f_{FeatureExtractor}()$ and $f_{Discriminator}()$. That is

$$f : x \in \mathbb{R}^{H\times W\times C} \rightarrow y \in U^K; \qquad f_{Discriminator}(f_{FeatureExtractor}(x)) = p(y|x)\ , \tag{1}$$

with predicted class label $y = \arg\max_k(p_k(y|x))$ and predicted confidence $c = \max_k(p_k(y|x))$ for $k = 1, ..., K$. A single CNN is referred to as the **baseline**.

3.2. Monte Carlo dropout

MCD samples $\hat{x} = f_{FeatureExtractor}(x)$ with masks drawn from a predefined distribution. Assuming sampling with masks drawn from the cascade of Gaussian and Bernoulli distributions, MCD samples $\hat{x}$ as

$$x^s = \hat{x} * \alpha^s * \beta^s, \quad with \quad \alpha_i^s \sim \mathcal{N}(1, \sigma^2 = \frac{q}{1-q}) \quad and \quad \beta_i^s \sim Bernoulli(q), \tag{2}$$

where q, α_i^s and β_i^s denote the dropout probability, and the elements of the sampled masks α^s and β^s, respectively. MCD estimates $\overline{p}(y|x)$ using the mean of S features sampling operations. That is,

$$\overline{p}(y|x) \approx \frac{1}{S}\sum_{s=1}^{S} p^s(y|x) \approx \frac{1}{S}\sum_{s=1}^{S} f_{Discriminator}(\hat{x}^s). \tag{3}$$

MCD is referred to as the average of S stochastic CNNs. Its main drawback is that it is sensitive to the sampling mask drawn from a predefined distribution parameterized by a dropout probability q, which is sensitive to the dataset and/or architecture [16].

3.3. Ensemble

Given a set of CNNs f_m for $m \in 1, 2, ..., M$, the ensemble prediction $\overline{p}(y|x)$ is estimated by averaging over the predictions of all CNNs. That is,

$$\overline{p}(y|x) := \frac{1}{M}\sum_{m=1}^{M} p^m(y|x) := \frac{1}{M}\sum_{m=1}^{M} f_{Discriminator_m}(f_{FeatureExtractor_m}(x)). \tag{4}$$

An ensemble is referred to as the average of M deterministic CNNs ($M << S$). Its major drawback is the inability to evaluate the uncertainty associated with the extracted features.

3.4. Mixture of Monte Carlo dropout

MMCD overcomes the drawback of an ensemble by applying MCD to individual ensemble members to evaluate the uncertainty associated with the extracted features. Given an ensemble, MMCD estimates $\overline{p}(y|x)$ as

$$\overline{p}(y|x) \approx \frac{1}{M \cdot S}\sum_{m=1}^{M}\sum_{s=1}^{S} p^{m_s}(y|x) \approx \frac{1}{M \cdot S}\sum_{m=1}^{M}\sum_{s=1}^{S} f_{Discriminator_m}(\hat{x}^{m_s}), \tag{5}$$

where $\hat{x}^{m_s}$ is a feature sampled (as shown in (2)) from $\hat{x}^m = f_{FeatureExtractor_m}(x)$. MMCD is referred to as the average of $M \cdot S$ stochastic CNNs. It has a similar drawback as MCD.

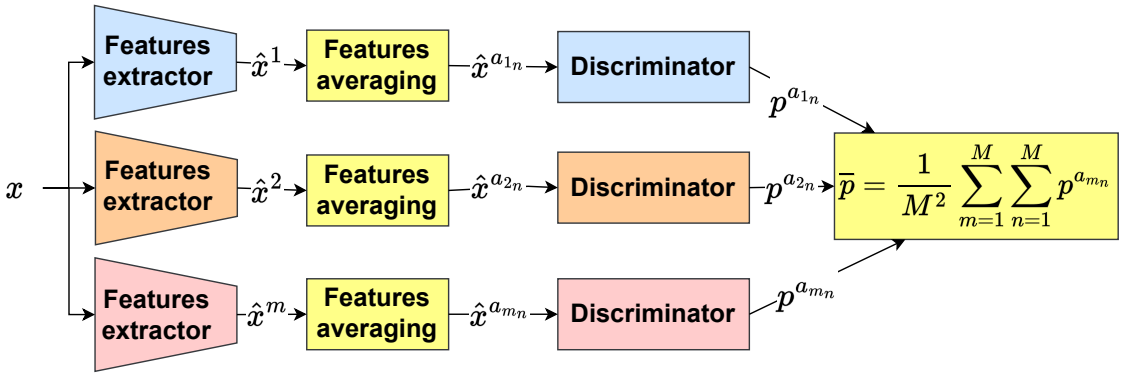


Figure 1. Overview of prediction estimation via features averaging inherent in MCA.

4. Proposed method: Monte Carlo averaging

Although the MMCD overcomes the drawback of an ensemble, it has drawbacks similar to MCD. To overcome this drawback, we proposed MCA, which replaces all feature sampling operations (inherent in MMCD) with averaging operations (see Figure 1) for feature perturbation. Particularly, MCA evaluates the uncertainty associated with the features extracted from an ensemble member $m \in 1, 2, ..., M$ by averaging (or perturbing) them sequentially with the features extracted from other ensemble members $n \in 1, 2, ..., M$. This is motivated by the hypothesis that features extracted from different ensemble members are different. To verify this hypothesis, we evaluated the classification accuracy when the discriminators of members m evaluate only features extracted from other members n, where $n \neq m$. That is, $\overline{p}(y|x)$ is estimated as

$$\overline{p}(y|x) \quad := \quad \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M} p^{m_n}(y|x) \quad := \quad \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M} f_{Discriminator_m}(\hat{x}^n)\ . \tag{6}$$

We found that with this, the classification accuracy drops drastically. For example, the classification accuracy of an ensemble trained on CIFAR10 dropped from 89.50% to 17.73% when we estimated $\overline{p}(y|x)$ as shown in (6). This means that the discriminators of members m cannot correctly classify the features extracted from other members n. This proves that *the features extracted from different ensemble members are different.* Therefore, MCA perturbs the features $\hat{x}_m$ (extracted from member m) by averaging them sequentially with the features $\hat{x}_n$ (extracted from other members n). That is, given a set of CNNs f_m, MCA estimates $\overline{p}(y|x)$ as

$$\overline{p}(y|x) \quad := \quad \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M} p^{a_{m_n}}(y|x) \quad := \quad \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M} f_{Discriminator_m}(\hat{x}^{a_{m_n}}), \tag{7}$$

where $\hat{x}^{a_{m_n}} = \frac{1}{2}\hat{x}^m + \frac{1}{2}\hat{x}^n$. Here, m can be equal to n to preserve the classification accuracy. MCA is referred to as the average of M^2 deterministic CNNs. Overall, the proposed MCA is an alternative to MMCD. Both MCA and MMCD have the same purpose and underlying principle. Specifically, both approaches evaluate multiple features extracted from multiple members and evaluate the uncertainty associated with the extracted features based on feature averaging or sampling.

5. Experimental setup

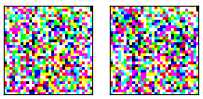
Training details Performance was expected to be dependent on task difficulty (dataset). This is because some datasets (e.g., GTSRB [36]) have more noise in their samples than others (e.g., MNIST [37]). Additionally, some datasets (e.g., CIFAR10 [38]) are more challenging to learn than others (e.g., MNIST [37]). Performance was also expected to be dependent on architecture because architecture determines how information is propagated from the input to subsequent layers and different architectures can result in different gradient computations and, thus, different solutions. Therefore, we compared MCA and related methods using four different datasets to evaluate their abilities to perform on different tasks with different difficulties. We compared these methods using three different architectures to evaluate their abilities to perform on different architectures. Specifically, we evaluated MNIST on VGGNets [1], FashionMNIST [39] on ResNets [2], CIFAR10 on DenseNets [3], and GTSRB on ResNets [2]. All CNNs were randomly initialized and trained with a random shuffling of training samples. All CNNs were trained with categorical cross entropy and stochastic gradient descent with a momentum of 0.9, a learning rate of 0.02, a batch size of 128, and epochs of 100. All CNNs were regularized with batch normalization [40] layers placed before each convolutional activation function and dropout layers placed at the inputs of the fully-connected layers. Regularization was also conducted using

standard data augmentation, such as rotation, translation, scaling, and shear. All images were standardized and normalized by dividing the pixel values by 255.

Inference details We applied features sampling in MCD and features averaging in MCA at inputs to the first fully-connected layers. MCD samples feature using masks drawn from a cascade of Bernoulli and Gaussian distributions [16] and using a dropout probability of 0.5. MCD samples 100 features ($S = 100$). Ensembles and MCA include 20 CNNs ($M = 20$).

(a) Test data

(b) Swap data



(c) Noisy data

Figure 2. Examples of evaluation data for CIFAR10.

Evaluation metrics We expect MCA and related methods to preserve the classification accuracy, while providing a good estimate of confidence. Therefore, we compared these methods with respect to the classification accuracy and quality of confidence. The quality of confidence was assessed by the degree of confidence calibration and the ability to separate TPs and FPs. The ability to separate TPs and FPs was assessed by evaluating the average confidence. The degree of the calibration was assessed by evaluating the calibration error using the expected calibration error (ECE) [41, 42], which is defined as shown in (10). It sorts and groups the predictions of evaluation data of size N into B equally-spaced bins and weighs the difference between the classification accuracy and the average confidence of the bins. The bin b_m denotes the set of indices of the evaluation sample t whose confidence falls into the interval $I_m =]\frac{m-1}{B}, \frac{m}{B}]$. The expected accuracy $acc(b_m)$ of bin b_m is estimated as in (9). The expected confidence $conf(b_m)$ within bin b_m is estimated as in (8). Confidence values are well-calibrated when $acc(b_m) = conf(b_m)\ \forall\ m \in [1; B]$.

$$conf(b_m) = \frac{1}{|b_m|} \sum_{t \in b_m} \hat{c}_t \tag{8}$$

$$acc(b_m) = \frac{1}{|b_m|} \sum_{t \in b_m} \mathbb{1}(\hat{y}_t = y_t) \tag{9}$$

$$ECE = \sum_{m=1}^{B} \frac{|b_m|}{N} |conf(b_m) - acc(b_m)| \tag{10}$$

Evaluation data We used five evaluation data (*test data*, *subsets of correctly classified test data*, *OOD data*, *swap data*, and *noisy data*) for different purposes. The *test data* were used to evaluate the classification accuracy and the ECE. We expect the classification accuracy to be high and the ECE to be low for *test data*. *Subsets of correctly classified test data* include 1000 correctly classified test data and were used to evaluate the average confidence for TPs. *Swap data* were simulated with *subsets of correctly classified test data* that were structurally perturbed by dividing the images into four regions and swapping the regions diagonally (see 2b). The *swap data* were used to evaluate the average confidence for FPs caused by structurally perturbed objects. *Noisy data* were simulated with *subsets of correctly classified test data* perturbed by additive Gaussian noise with a standard deviation of 500 (see 2c). The *noisy data* were used to evaluate the average confidence for FPs caused by noisy objects. *OOD data* were simulated using 1000 test data from CIFAR100 [38] and were used to evaluate the average confidence on FPs caused by unknown objects. TPs and FPs are separable when the confidence for TPs is high and the confidence for FPs is low. Therefore, we expect the average confidence to be high for TPs and low for FPs.

Table 1. Summary of classification accuracy (CA), average confidence (AC), and expected calibration error (ECE) obtained on *test data*.

	CA (AC) [%]	**ECE** [10^{-2}]
CIFAR10 (DenseNets)		
Baseline	86.02 (86.36)	**2.06**
MCD	85.65 (70.57)	15.11
Ensemble	90.15 (83.48)	6.75
MMCD	89.70 (69.84)	19.87
MCA	89.75 (69.27)	20.49
FashionMNIST (ResNets)		
Baseline	90.23 (89.76)	**1.88**
MCD	90.32 (79.00)	11.41
Ensemble	92.99 (86.87)	6.34
MMCD	93.10 (75.63)	17.48
MCA	92.96 (69.88)	23.09
MNIST (VGGNets)		
Baseline	98.92 (98.48)	**0.70**
MCD	98.96 (95.01)	3.99
Ensemble	99.11 (98.25)	1.12
MMCD	99.13 (94.78)	4.48
MCA	99.13 (85.34)	13.81
GTSRB (ResNets)		
Baseline	93.41 (97.17)	3.87
MCD	93.38 (90.30)	3.54
Ensemble	94.68 (92.55)	**2.59**
MMCD	94.31 (86.71)	7.73
MCA	94.62 (77.56)	17.18

Table 2. Summary of average confidence (AC) [%] obtained on true predictions (TPs) (↑) and false predictions (FPs) (↓): TPs were obtained on *subsets of correctly classified test data*. FPs were obtained on *swap*, *noisy*, or *OOD* data.

	TPs	**FPs (OOD)**	**FPs (Swap)**	**FPs (Noisy)**
CIFAR10 (DenseNets)				
Baseline	**95.94**	88.29	57.88	80.48
MCD	82.79	35.72	38.56	33.01
Ensemble	96.40	38.41	50.28	56.10
MMCD	83.48	24.34	**35.63**	**28.92**
MCA	83.20	**19.40**	36.17	30.76
FashionMNIST (ResNets)				
Baseline	**96.18**	54.69	73.93	91.35
MCD	85.66	35.69	56.57	81.48
Ensemble	95.06	49.83	55.71	58.49
MMCD	83.27	37.40	44.21	39.15
MCA	77.12	**32.83**	**37.83**	**35.99**
MNIST (VGGNets)				
Baseline	**99.38**	60.86	61.58	97.37
MCD	96.00	44.93	49.39	88.79
Ensemble	99.34	55.95	52.59	62.93
MMCD	95.88	48.23	43.02	58.45
MCA	86.40	**38.88**	**34.91**	**47.45**
GTSRB (ResNets)				
Baseline	**99.79**	56.87	53.63	50.21
MCD	94.53	26.14	31.19	27.57
Ensemble	99.20	34.12	39.98	29.96
MMCD	92.61	17.92	28.13	21.34
MCA	84.87	**16.14**	**21.69**	**17.76**

6. Experimental results

Comparison of classification accuracy and calibration error of MCA and related methods Table 1 summarizes the classification accuracy, average confidence, and ECE for test data. The results show that *MCD can preserve the classification accuracy of the baseline (single CNN), since the increase/decrease in classification accuracy of the baseline caused by MCD is minimal.* For example, for CIFAR10, MCD decreases the classification accuracy of the baseline from 86.02% to (only) 85.65%. Moreover, for FashionMNIST, MCD increases the classification accuracy of the baseline from 90.23% to (only) 90.32%. Furthermore, the results show that *an ensemble can increase the classification accuracy of the baseline, since the increase in classification accuracy of the baseline caused by an ensemble is significant.* For example, for CIFAR10, the ensemble increases the classification accuracy of the baseline from 86.02% to 90.15%. However, the results show that *MMCD and MCA can preserve the classification accuracy of the underlying ensemble, since the increase/decrease in classification accuracy of an*

ensemble caused by MMCD or MCA is minimal. Table 1 shows that the ECE of a baseline is lower than that of an ensemble, which is in turn lower than that of MCD, MMCD, and MCA. This means that *baseline is better calibrated than ensemble, which is in turn better calibrated than MCD, MMCD, and MCA*. Table 1 also shows that the ECE of MCD is lower than that of MMCD, which is in turn lower than that of MCA. This means that, *MCD is better calibrated than MMCD, which is in turn better calibrated than MCA*.

Comparison of the ability of MCA and related methods to separate TPs and FPs Table 2 summarizes the average confidence (AC) for TPs and FPs. The results show that *ensembles maintain high confidence for TPs like baselines.* However, *MCD, MMCD, and MCA reduce the confidence of TPs.* For example, for FashionMNIST, ensemble reduces the AC of baseline for TPs from 96.18% to (only) 95.06%, while MCD, MMCD, and MCA reduce it to 85.66%, 83.27%, and 77.12%, respectively. Furthermore, the results show that the AC of baseline is larger than that of other methods for all FPs. This means that *MCD, ensemble, MMCD, and MCA reduce the confidence of baseline for FPs.* However, *the ability of the ensemble to reduce the confidence of FPs better than MCD is dependent on the dataset, architecture, or type of FPs.* For example, for FashionMNIST, the ensemble reduces the AC of baseline for FPs from 91.35% to 58.49% due to noisy data, whereas the MCD reduces it to (only) 81.48%. By contrast, for FPs due to the OOD data, the ensemble reduces the AC of the baseline from 54.69% to (only) 49.83%, whereas the MCD reduces it to 35.69%. However, for CIFAR10, the AC of the ensemble for all FPs is higher than that of MCD. Furthermore, Table 2 shows that *MMCD and MCA reduce the confidence of the underlying ensemble for all FPs.* Table 2 also shows that *MCA can maintain low confidence for FPs similar to or sometimes even better than MMCD.*

Table 3. Mean and standard deviation of inference time (in seconds) obtained over 100 test samples of CIFAR10 evaluated on DenseNets.

	Inference time[s]
Baseline	0.06 ± 0.01
MCD	1.33 ± 0.09
Ensemble	1.22 ± 0.05
MMCD	22.30 ± 0.42
MCA	5.00 ± 0.25

Comparison of the inference time of MCA and related methods Table 3 shows that MMCD is more than four times more expensive than MCA, which is also more expensive than an ensemble and MCD. This means that both MMCD and MCA increase the inference time.

Assessing the design benefit of MCA over MMCD We reduced the capacity of DenseNets by reducing the number of parameters from 21.36 million to 5.02 million and retrained them on CIFAR10. We then evaluated classification accuracy, AC, and ECE for test data (see Table 4). The results show that the ECE of MCA is footnotesizeer than that of MMCD. This indicates that the confidence drop on TPs is larger for MMCD than for MCA. Here, MMCD fails because the predefined distribution from which the masks were drawn was not fine-tuned when the capacity of the DenseNets was reduced. Particularly, the dropout probability of 0.5 is too large for DenseNets with a footnotesize capacity.

Table 4. Summary of classification accuracy (CA), average confidence (AC), and expected calibration error (ECE) obtained on CIFAR10 evaluated on DenseNets with footnotesize capacity.

	CA (AC) [%]	**ECE** [10^{-2}]
MMCD	86.67 (57.58)	29.10
MCA	89.03 (68.04)	21.01

7. Discussion

To achieve our goal of improving the separability of TPs and FPs by enforcing the confidence to be high for TPs and low for FPs, the research question *What method is required to achieve high and low confidence for TPs and FPs, respectively?* must be answered. To address this question, MCA was proposed and compared to related methods (baseline (single CNN), MCD, ensemble, and MMCD). We showed that MCD could preserve the accuracy of the baseline, while reducing the confidence for TPs. This finding indicates that MCD (mainly) affects the degree of confidence than accuracy. Conversely, we showed that an ensemble can increase the accuracy of the baseline, while maintaining high confidence for TPs. This result is consistent with previous studies [1, 2] showing that ensemble can increase accuracy. This is because an ensemble evaluates multiple features. We showed that MMCD and MCA can preserve the accuracy of the underlying ensemble, while reducing the confidence for TPs. However, this result suggests that, similar to MCD/MMCD, MCA (mainly) affects the degree of confidence than accuracy. This is because feature sampling in MCD/MMCD and feature averaging in MCA evaluate the uncertainty associated with a given feature, but does not change the prediction associated with the feature. We showed that baseline is (often) better calibrated than ensemble, which is (often) better calibrated than MCD, MMCD, and MCA. Moreover, we showed that MCD is (often) better calibrated than MMCD, which is (often) better calibrated than MCA. This is because MCD, MMCD, and MCA reduce the confidence of TPs. The larger the decrease in the degree of confidence for TPs, the larger the calibration error. We showed that ensemble can reduce the confidence of baseline for FPs, while maintaining the confidence for TPs (nearly) unchanged. However, MCD, MMCD, and MCA can reduce the confidence of baseline for FPs at the cost of reducing the confidence for TPs. We showed that the ability of an ensemble to reduce the confidence of FPs better than MCD is dependent on the dataset, architecture, or FP type. This result suggests that we cannot claim that the ensemble performs better than MCD in terms of capturing uncertainty. However, this contradicts previous studies [25, 26, 9], which claim that ensemble captures uncertainty better than MCD. Although MMCD and MCA reduce the confidence for TPs, the remaining confidence for TPs is still high ($[50\%, 100\%]$) whereas the confidence for FPs is low ($[0\%, 50\%[$). Therefore, we hypothesized that MCA and MMCD can separate TPs and FPs better than ensemble or MCD. This is because MCA and MMCD not only evaluate multiple features extracted from different members like an ensemble, but also evaluate the uncertainty associated with the extracted features. For this reason, MCA and MMCD capture the diversity between the different members better than an ensemble and therefore improve the uncertainty. We showed that MCA can maintain low confidence for FPs similar to or sometimes even better than MMCD. Hence, we hypothesized that MCA can perform (in terms of separating TPs and FPs) similar to or sometimes even better than MMCD. Although MMCD and MCA have similar performance, the design process of MMCD is more complex than that of MCA. This is because MMCD requires the specification of a prior distribution from which masks will be drawn for feature sampling, whereas MCA relies on features extracted from ensemble members. Besides, MMCD is more expensive than MCA because of the large number of sampling operations.

8. Conclusion

By sequentially averaging the features of ensemble members, MCA evaluates the uncertainty associated with the extracted features like MMCD. Based on the empirical comparison of MCA and related methods, we conclude that MCA can obtain performance similar to or sometimes even better than MMCD. Particularly, like MMCD, MCA can preserve the accuracy of the underlying ensemble. MCA, like MMCD, can separate TPs and FPs better than baseline, ensemble, and MCD. This finding suggests that we can use MCA instead of MMCD for applications (such as collision prediction [10]), where the separability of TPs and FPs is

essential. MCA can also benefit other fields (such as active learning [17], online learning [25], and reinforcement learning [33]) where uncertainty is required.

9. Limitations

Although MCA can improve the separability of TPs and FPs, it can increase the calibration error because it reduces the confidence in TPs. This suggests that improving the separability of TPs and FPs may negatively affect confidence calibration, and vice versa. We argue that the confidence drop in TPs is caused by inductive biases inherent in ensemble members or introduced by feature averaging. To reduce the level of inductive biases, we can combine ensemble members by averaging logits instead of probabilities. This will, however, be investigated in future works. MCA relies on multiple members like ensemble and MMCD, and for a large number of members, it may require a large amount of storage memory. This may limit its adoption in applications with a limited amount of storage memory. To overcome this limitation, future research should explore pruning methods [43] to reduce the number of members to three or five and therefore reduce the memory requirement.

References

[1] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014.

[2] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770-8.

[3] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 4700-8.

[4] Bejani MM, Ghatee M. A systematic review on overfitting control in shallow and deep neural networks. Artificial Intelligence Review. 2021;54(8):6391-438.

[5] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. arXiv preprint arXiv:161201474. 2016.

[6] Hendrycks D, Dietterich T. Benchmarking neural network robustness to common corruptions and perturbations. arXiv preprint arXiv:190312261. 2019.

[7] Hendrycks D, Gimpel K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In: 5th International Conference on Learning Representations; 2017. .

[8] Liang S, Li Y, Srikant R. Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:170602690. 2017.

[9] Ovadia Y, Fertig E, Ren J, Nado Z, Sculley D, Nowozin S, et al. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In: Advances in Neural Information Processing Systems; 2019. p. 13991-4002.

[10] Kahn G, Villaflor A, Pong V, Abbeel P, Levine S. Uncertainty-aware reinforcement learning for collision avoidance. arXiv preprint arXiv:170201182. 2017.

[11] Chen W, Qu T, Zhou Y, Weng K, Wang G, Fu G. Door recognition and deep learning algorithm for visual based robot navigation. In: 2014 ieee international conference on robotics and biomimetics (robio 2014). IEEE; 2014. p. 1793-8.

[12] Ouyang W, Wang X. Joint deep learning for pedestrian detection. In: Proceedings of the IEEE international conference on computer vision; 2013. p. 2056-63.

[13] Gawlikowski J, Tassi CRN, Ali M, Lee J, Humt M, Feng J, et al. A survey of uncertainty in deep neural networks. arXiv preprint arXiv:210703342. 2021.

[14] Gal Y, Ghahramani Z. Bayesian convolutional neural networks with Bernoulli approximate variational inference. arXiv preprint arXiv:150602158. 2015.

[15] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: international conference on machine learning; 2016. p. 1050-9.

[16] Tassi CRN. Bayesian Convolutional Neural Network: Robustly Quantify Uncertainty for Misclassifications Detection. In: Mediterranean Conference on Pattern Recognition and Artificial Intelligence. Springer; 2019. p. 118-32.

[17] Zeng J, Lesnikowski A, Alvarez JM. The relevance of Bayesian layer positioning to model uncertainty in deep Bayesian active learning. arXiv preprint arXiv:181112535. 2018.

[18] Brosse N, Riquelme C, Martin A, Gelly S, Moulines É. On last-layer algorithms for classification: Decoupling representation from uncertainty estimation. arXiv preprint arXiv:200108049. 2020.

[19] Kristiadi A, Hein M, Hennig P. Being bayesian, even just a bit, fixes overconfidence in relu networks. In: International Conference on Machine Learning. PMLR; 2020. p. 5436-46.

[20] Mobiny A, Nguyen HV, Moulik S, Garg N, Wu CC. DropConnect Is Effective in Modeling Uncertainty of Bayesian Deep Networks. arXiv preprint arXiv:190604569. 2019.

[21] McClure P, Kriegeskorte N. Robustly representing uncertainty through sampling in deep neural networks. arXiv preprint arXiv:161101639. 2016.

[22] Zhang Z, Dalca AV, Sabuncu MR. Confidence calibration for convolutional neural networks using structured dropout. arXiv preprint arXiv:190609551. 2019.

[23] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012;25.

[24] Ashukha A, Lyzhov A, Molchanov D, Vetrov D. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. arXiv preprint arXiv:200206470. 2020.

[25] Beluch WH, Genewein T, Nürnberger A, Köhler JM. The power of ensembles for active learning in image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 9368-77.

[26] Gustafsson FK, Danelljan M, Schon TB. Evaluating scalable bayesian deep learning methods for robust computer vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops; 2020. p. 318-9.

[27] Lee S, Purushwalkam S, Cogswell M, Crandall D, Batra D. Why M heads are better than one: Training a diverse ensemble of deep networks. arXiv preprint arXiv:151106314. 2015.

[28] Wen Y, Jerfel G, Muller R, Dusenberry MW, Snoek J, Lakshminarayanan B, et al. Combining Ensembles and Data Augmentation Can Harm Your Calibration. In: International Conference on Learning Representations; 2021. .

[29] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations; 2018. .

[30] Maroñas J, Ramos D, Paredes R. Improving Calibration in Mixup-trained Deep Neural Networks through Confidence-Based Loss Functions. arXiv preprint arXiv:200309946. 2020.

[31] Rahaman R, Thiery AH. Uncertainty Quantification and Deep Ensembles. stat. 2020;1050:20.

[32] Wu X, Gales M. Should Ensemble Members Be Calibrated? arXiv preprint arXiv:210105397. 2021.

[33] Lütjens B, Everett M, How JP. Safe reinforcement learning with model uncertainty estimates. In: 2019 International Conference on Robotics and Automation (ICRA). IEEE; 2019. p. 8662-8.

[34] Wilson AG, Izmailov P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, editors. Advances in Neural Information Processing Systems. vol. 33; 2020. p. 4697-708.

[35] Fort S, Hu H, Lakshminarayanan B. Deep ensembles: A loss landscape perspective. arXiv preprint arXiv:191202757. 2019.

[36] Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In: The 2013 international joint conference on neural networks (IJCNN). Ieee; 2013. p. 1-8.

[37] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;86(11):2278-324.

[38] Krizhevsky A, Hinton G, et al. Learning multiple layers of features from tiny images. 2009.

[39] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:170807747. 2017.

[40] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR; 2015. p. 448-56.

[41] Naeini MP, Cooper GF, Hauskrecht M. Obtaining well calibrated probabilities using bayesian binning. In: Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence. vol. 2015. NIH Public Access; 2015. p. 2901.

[42] Guo C, Pleiss G, Sun Y, Weinberger KQ. On calibration of modern neural networks. In: International Conference on Machine Learning. PMLR; 2017. p. 1321-30.

[43] Tsoumakas G, Partalas I, Vlahavas I. A taxonomy and short review of ensemble selection. In: Workshop on Supervised and Unsupervised Ensemble Methods and Their Applications; 2008. p. 1-6.