



# A novel uncertainty-aware deep learning technique with an application on skin cancer diagnosis

Afshar Shamsi<sup>1,6</sup> · Hamzeh Asgharnezhad<sup>5</sup> · Ziba Bouchani<sup>2</sup> · Khadijeh Jahanian<sup>3</sup> · Morteza Saberi<sup>3</sup> · Xianzhi Wang<sup>3</sup> · Imran Razzak<sup>4</sup> · Roohallah Alizadehsani<sup>5</sup> · Arash Mohammadi<sup>6</sup> · Hamid Alinejad-Rokny<sup>1,7,8</sup>

Received: 9 June 2022 / Accepted: 24 July 2023 / Published online: 19 August 2023  
© Crown 2023

## Abstract

Skin cancer, primarily resulting from the abnormal growth of skin cells, is among the most common cancer types. In recent decades, the incidence of skin cancer cases worldwide has risen significantly (one in every three newly diagnosed cancer cases is a skin cancer). Such an increase can be attributed to changes in our social and lifestyle habits coupled with devastating man-made alterations to the global ecosystem. Despite such a notable increase, diagnosis of skin cancer is still challenging, which becomes critical as its early detection is crucial for increasing the overall survival rate. This calls for advancements of innovative computer-aided systems to assist medical experts with their decision making. In this context, there has been a recent surge of interest in machine learning (ML), in particular, deep neural networks (DNNs), to provide complementary assistance to expert physicians. While DNNs have a high processing capacity far beyond that of human experts, their outputs are deterministic, i.e., providing estimates without prediction confidence. Therefore, it is of paramount importance to develop DNNs with uncertainty-awareness to provide confidence in their predictions. Monte Carlo dropout (MCD) is vastly used for uncertainty quantification; however, MCD suffers from overconfidence and being miss calibrated. In this paper, we use MCD algorithm to develop an uncertainty-aware DNN that assigns high predictive entropy to erroneous predictions and enable the model to optimize the hyper-parameters during training, which leads to more accurate uncertainty quantification. We use two synthetic (two moons and blobs) and a real dataset (skin cancer) to validate our algorithm. Our experiments on these datasets prove effectiveness of our approach in quantifying reliable uncertainty. Our method achieved  $85.65 \pm 0.18$  prediction accuracy,  $83.03 \pm 0.25$  uncertainty accuracy, and  $1.93 \pm 0.3$  expected calibration error outperforming vanilla MCD and MCD with loss enhanced based on predicted entropy.

**Keywords** Deep learning · Machine learning · Classification · Uncertainty quantification · Skin cancer

## 1 Introduction

Over the past decades, there has been a significant increase in the incidence of skin cancers across the world potentially due to our social and lifestyle changes, as well as the depletion of the ozone layer [1]. Skin cancer, which is mostly caused by abnormal growth of skin cells, can be considered as one of the most prevalent cancer types. One in six Americans suffer from skin cancer at some point in their lives. This type of cancer accounts for one-third of all cancers in the United States. Near 75 percent of all skin cancer-related deaths are due to malignant melanoma. The most common type of skin cancer is non-melanoma with

lower mortality rate. Despite the aforementioned increase in incidence of skin cancer, its diagnose is still challenging even for dermatologists [2–4]. Consequently, there has been a surge of interest in incorporation of computer-aided methodologies to assist the medical experts with their decision makings [5]. Recently, machine learning (ML), in particular, deep learning (DL) solutions, have achieved promising results in various application domains, encouraging their utilization for cancer screening/diagnosis [6]. Capitalizing on the fact that early diagnosis of skin cancer is of significant importance to improve life expectancy of patients, researchers strive to develop advanced DL models in this domain.

Generally speaking, DL methods have become widely popular for medical image processing and analysis tasks

Extended author information available on the last page of the article

such as segmentation of cancerous lesions. For instance, convolutional neural network (CNN) has been recently used for systematic categorization of skin lesion diseases [7] and its performance has even been evaluated against a group of 58 dermatologists for skin cancer classification. As pointed out in [8], in another similar attempt, promising results have been obtained by using CNN for skin cancer classification [9–11]. The resulting DL model can be used as a complementary medical assistant for skin cancer classification [12]. One of the milestones of applying DL in medical applications is the lack of labeled data. This stems from the fact that labeling of medical images must be done by experts, which is both costly and time consuming. Additionally, deep neural networks (DNNs), typically, have thousands of parameters, therefore, learning appropriate parameter values from scratch may be computationally demanding. An alternative approach is transfer learning, which is generally used to fine-tune a pre-trained network to a new task and achieve acceptable performance in reasonable amount of time [13]. Deep neural networks (DNNs) have shown promising results in challenging tasks; however, their reliability is fragile when it comes to unseen data, which could lead to wrong medical diagnoses and, ultimately, patient deaths. This is due to the fact that DNNs are black-box models with deterministic behavior, lacking transparency in their decision-making process and the ability to estimate how certain they are about their own predictions. Several studies have been devoted to the important topic of uncertainty quantification for deep models [14].

Generally speaking, uncertainty is of two types: epistemic and aleatoric. Epistemic uncertainty, also known as model uncertainty, is due to limited training data or model complexity, which is reducible by providing enough data. Aleatoric uncertainty, on the other hand, arises due to the inherent noise of observations, making it data-dependent and cannot be reduced by gathering more data [?].

One possible way of uncertainty quantification in DNNs is through the Bayesian approach. Instead of using deterministic values for DNN parameters, the Bayesian approach imposes a probability distribution on them, providing a natural way of capturing uncertainty [15]. However, the computational complexity of this approach is high, hindering the estimation of posterior distribution in real time [13]. Bayesian neural networks begin with an initial value on the previous model and data parameters and use them to calculate and update the posterior distribution. For networks with thousands of parameters, finding and calculating the posterior distribution is complex and computationally costly. Estimation methods such as Monte Carlo (MC) can be used to address the computational complexity of the Bayesian approach. In this method, each sample is fed  $M$  times to the network equipped with

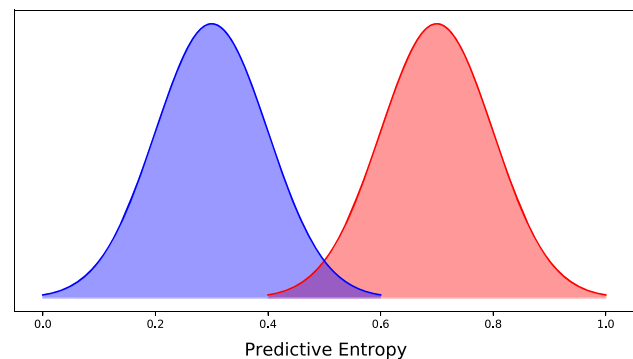
dropout layers. Random activation/deactivation of the network neurons (due to dropout layers) yields  $M$  different outputs for the given input. These  $M$  outputs can be used to estimate mean and variance, providing a measure of uncertainty. However, the drawback of MC is the lack of calibration of forecasts, leading to performance lower than ensemble methods [16].

The purpose of this paper is to propose a new method for improving the quantification of uncertainty in skin cancer detection models. Ideally, the model should assign high uncertainty to predictions that are not certain. By sorting the model's outcomes according to their predictive entropy, we can estimate two distributions that correspond to correct and incorrect predictions, as shown in Fig. 1.

In our previous work [16], we addressed the drawbacks of the MCD method, including overconfidence and being non-calibrated, by proposing an uncertainty-aware loss function to optimize the model parameters. Although the use of this loss function leads to uncertainty quantification with acceptable accuracy, optimizing the model's hyperparameters still requires manual intervention. Suboptimal choices of hyperparameter values can impede further improvements in MCD.

Inspired by the fact that MCD is very sensitive to the choice of hyperparameters and the issues discussed above, the main contribution of this paper can be listed as:

- In addition to optimizing uncertainty accuracy (UA) and expected calibration error (ECE), our previously proposed uncertainty-aware loss function is exploited for automatic hyperparameter (e.g. dropout probability) tuning based on Bayesian optimization.
- For the first time, uncertainty-aware diagnosis of skin cancer is investigated. Given that field of medical diagnosis is safety critical, robust uncertainty quantification is vital to successful deployment of DNNs in this field.



**Fig. 1** The density plots of predictive uncertainty predictions for correctly classified and misclassified samples of the ideal classification model

- While Ensemble methods outperform MCD in terms of UA and ECE, they are computationally expensive. Achieving better uncertainty quantification but keeping computation complexity low is highly desirable. To this end, MCD is reinforced with better UA and ECE estimation as well as automatic hyperparameter tuning while computational complexity is much lower than Ensemble methods.

Through comprehensive experiments, we quantitatively evaluate our method against the existing ones such as the MCD and the Ensemble Bayesian Networks using cross-validation. We also perform a qualitative evaluation on the distributions of correctly and incorrectly categorized predictions, due to the fact that the ideal model has to separate the two distributions (their means should be as far as possible) and reduce their overlap. As a result, our model is able to detect wrong predictions which can be sent to a medical expert for further inspection.

The rest of the article is organized as follows: the background of our work is introduced in Sect. 2. The proposed method for better uncertainty quantification is presented in Sect. 3. The simulation and experimental results of our proposed method are given in Sect. 4. Finally, the paper is closed with conclusion in Sect. 5.

## 2 Background

In uncertainty-aware classification, two types of accuracy can be considered which are related to model prediction and model uncertainty quantification [?]. The motivation for uncertainty quantification is the fact that deterministic models are bound to predict the class for given input even if samples similar to it have not been seen during training. Under such circumstances, performing a prediction may lead to erroneous results. On the other hand, uncertainty-aware models are able to provide a measure of how certain they are about their own prediction. This way, the user will know when it is safe to trust the model’s prediction. However, solely providing an estimation of model uncertainty will not be of much use. It is vital to compute the accuracy of the model’s uncertainty estimation. A model with high uncertainty estimation accuracy and a high classification accuracy is the ideal case. Such a model is expected to assign a low uncertainty to its correct predictions and high uncertainty to its incorrect ones [17]. In the remainder of this section, the required background for uncertainty quantification is reviewed.

## 2.1 Predictive uncertainty quantification

For a given test sample, the predicted output of a model is either correct or incorrect. Using multiple forward passes of MCD, the predictive mean and standard deviation for the test sample can be computed. If the predictive standard deviation is less than a certain threshold, the model prediction is considered to be certain. For standard deviation higher than the threshold, the model is assumed to be uncertain about its own prediction. Considering all possible combinations of {correct, incorrect} and {certain, uncertain} yields the confusion matrix shown in Fig. 2. In this matrix, TC represents the cases in which the model has made a correct prediction that it is certain about. TU stands for the cases that the model has made a wrong prediction and the model is highly uncertain about its prediction correctness. FU is related to the cases that the model is uncertain about its own correct predictions. Finally, FC represents the worst error type which occurs when the model makes a wrong prediction with high confidence. Similar to the idea of the confusion matrix, we can propose four metrics for evaluating different Bayesian networks:

- Uncertainty sensitivity ( $U_{sen}$ ) which denotes the number of incorrect and uncertain estimates divided by the number of incorrect predictions:

$$U_{sen} = \frac{TU}{TU + FC} \tag{1}$$

- Uncertainty Specificity ( $U_{spe}$ ) which is defined as the number of correct and certain predictions divided by the number of correct predictions:

$$U_{spe} = \frac{TC}{TC + FU} \tag{2}$$

- Uncertainty precision ( $U_{pre}$ ) which is related to the number of incorrect and uncertain predictions divided by the number of uncertain predictions:

$$U_{pre} = \frac{TU}{TU + FU} \tag{3}$$

- Uncertainty accuracy (UA) which is defined as the sum of diagonal predictions (TU and TC) divided by the total number of predictions:

		Confidence	
		Certain	Uncertain
Correctness	Correct	True Certainty (TC)	False Uncertainty (FU)
	Incorrect	False Certainty (FC)	True Uncertainty (TU)

Fig. 2 The UCM and its components

$$UA = \frac{TU + TC}{TU + TC + FU + FC} \tag{4}$$

A trustworthy model has a high UA. The best value of UA is 1 and the worst value is 0. A network with UA metric near one is able to estimate its uncertainty level more accurately and identify the samples about which it is uncertain.

## 2.2 Uncertainty quantification techniques

The two uncertainty prediction techniques used in Bayesian neural networks are briefly described in the following subsections.

nature of the dropout layers, the  $M$  predictions are not exactly the same and can be used to compute the mean and the standard deviation of the posterior distribution. The standard deviation is considered a measure of uncertainty. Higher standard deviation means higher uncertainty. More details on MCD method is already available in the literature [19]. The predictive mean ( $\mu_{pred}$ ) for a test input  $x$  is estimated as:

$$\mu_{pred}(x, c) \approx \frac{1}{M} \sum_{m=1}^M p(y = c | x, \hat{\omega}_m), \tag{5}$$

where  $p(y = c | x, \hat{\omega}_m)$  is the softmax probability of sample  $x$  belonging to class  $c$ ,  $\hat{\omega}_m$  is the set of network parameters in  $m^{th}$  forward pass, and  $M$  is the number of forwarding

---

### Algorithm 1 MCD

---

**Require:** Labeled data (Skin Cancer images)  $(x_i, y_i)_{i=1}^{10015}$ , #iterations  $T$ , #MCD forward passes  $M$   
**Ensure:** Finding PEs  
 1: **for**  $t = 1$  to  $T$  **do**  
 2:     Randomly split Dataset to train and test  
 3:     Train the model  
 4:     **for**  $m = 1$  to  $M$  **do**  
 5:         Perform forward pass with dropout  
 6:          $\hat{y}_M = model(x_M)$   
 7:     **end for**  
 8:     Calculate the PEs  
 9: **end for**

---

### 2.2.1 Monte Carlo dropout (MCD)

MCD is a Gaussian Process (GP) approximation that consists of retaining rate, model precision, and length scale parameters. According to [18], the posterior distribution of a Bayesian setting can be estimated by turning on the dropout at test time and feeding each sample to the network  $M$  times which yields  $M$  predictions. Due to the random

passes of MCD. For uncertainty evaluation, the predictive entropy (PE) is calculated as follows [20, 21]:

$$PE(x) = - \sum_{c=1}^C \mu_{pred}(x, c) \log [\mu_{pred}(x, c)] \tag{6}$$

where  $C$  is the number of classes. Pseudo code of generating MCD is revealed in Algorithm 1.

---

### Algorithm 2 Ensemble

---

**Require:** Labeled data (Skin Cancer images)  $(x_i, y_i)_{i=1}^{10015}$ , ensemble size  $N$   
 1: Randomly split Dataset to train and test  
 2: **for**  $i = 1$  to  $N$  **do** ▷ Ensemble creation  
 3:     Generate the random models  
 4: **end for**  
 5: Train the models in the ensemble using the training set  
 6: Get the models output on test set  
 7: Average the models outputs using equation 7  
 8: Calculate the PEs using equation 8

---

### 2.2.2 Ensemble Bayesian networks

Probabilistic Bayesian methods are accepted as one of the best ways to quantify uncertainty. However, ensemble methods are computationally expensive leading to proposal of many approximate solutions. As an example, ensemble of neural networks is exploited to quantify epistemic uncertainties. Ensemble of networks is useful provided that the networks parameters are different enough from each other [22].

There are two types of methods for generating ensemble networks. In the first approach, for each sample multiple neural networks provide their predictions for which mean and standard deviation are computed. The mean is the final prediction and the standard deviation represents the ensemble uncertainty about the computed mean. This method is called ensembling of the models. In the second approach, a network is trained on different subsets of data, which is known as ensembling the data. In this paper, the first approach is used. Using an ensemble of  $N$  networks, the probability that input sample  $x$  belongs to class  $c$  can be estimated as follows [23]:

$$\hat{p}(y = c | x) = \frac{1}{N} \sum_{i=1}^N p_{\theta_i}(y = c | x) \tag{7}$$

where  $c$  is the class index and  $\theta_i$  denotes the set of parameters for  $i^{th}$  network of the ensemble. Moreover, the PE metric is defined as:

$$PE(x) = - \sum_{c=1}^C \hat{p}(y = c | x) \log[\hat{p}(y = c | x)] \tag{8}$$

Algorithm 2 shows the pseudo code of the ensemble setting.

### 3 Proposed method

Despite being powerful learners, DNNs performance is directly affected by the right choice of hyperparameter values. Moreover, DNNs are deterministic in nature and incapable of quantifying uncertainty about their own predictions. As mentioned before, methods like MCD can be used to make DNNs uncertainty-aware. However, MCD is known to provide miscalibrated uncertainty estimates [23] which we have tackled by proposing an uncertainty-aware multi-objective loss function [16]:

$$\hat{\mu}_{b,c}^{pred} = \frac{1}{M} \sum_{m=1}^M y_{b,c}^{(m)}, b = 1 : B, c = 1 : C, \tag{9}$$

**Table 1** The resulted optimized parameters for each dataset by MCD+Entropy BO algorithm

Datasets	Parameters			
	L1	L2	P1	P2
Two moons	88	34	0.34	0.45
Blobs	91	35	0.25	0.47
Skin cancer	100	51	0.1	0.47

$$\hat{\mu}_b^{pred} = \operatorname{argmax}_c \hat{\mu}_{b,c}^{pred}, \tag{10}$$

$$Loss = \frac{1}{B} \sum_{b=1}^B \left\{ \underbrace{- \left[ y_b \log(\hat{\mu}_b^{pred}) + (1 - y_b) \log(1 - \hat{\mu}_b^{pred}) \right]}_{\text{Binary cross entropy}} + \sum_{m=1}^M PE(x_b^{(m)}) \right\}, \tag{11}$$

where  $M$  is the number of forward passes of MCD,  $B$  is the batch size,  $C$  is the number of classes,  $y_b$  is the desired label for input sample  $x_b$ ,  $\hat{y}_{b,c}^{(m)}$  is a  $[B \times C]$  matrix such that its  $b^{th}$  row is the network prediction (softmax output) corresponding to  $x_b$ . The left-hand side of equation 9 is a  $[B \times C]$  matrix and the left-hand side of equation 10 is a  $[B \times 1]$  vector due to applying argmax operator. Finally,  $PE(x_b^{(m)})$  is the predictive entropy for  $x_b$  in the  $m^{th}$  forward pass of MCD. The loss function in equation 11 is the sum of traditional cross entropy and PEs aiming to minimize the overlapped region depicted in Fig. 1. According to the definition of the PE uncertainty metric, lower/higher PE means that the network has higher/lower confidence in its prediction. Using the loss function in equation 11 leads to the uncertainty aware training approach called MCD+entropy which improves MCD in terms of providing uncertainty estimates with better calibration.

The pseudo-code of MCD+Entropy method is available in Algorithm 3. In line 2, the  $T$  training epochs are started. In each epoch, the training set is traversed batch by batch. For each batch  $M$ , forward passes are computed (line 5) which are used to compute the PEs in line 7. After computing the loss value (line 8), the network parameters are updated using backpropagation at line 9. At the end of each epoch, the evaluation of the model is done on the test set (line 11).



**Algorithm 3** MCD+entropy

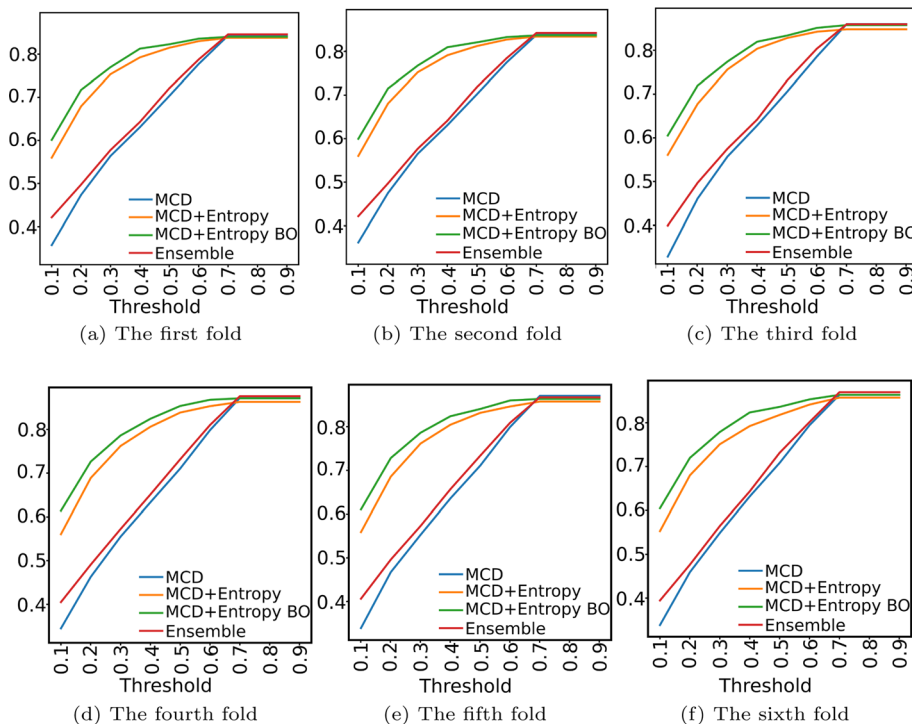
**Require:** Labeled data (Skin Cancer images)  $(x_i, y_i)_{i=1}^{10015}$ , batch size  $B$ , #epochs  $T$ , #MCD forward passes  $M$

- 1: Randomly split Dataset to train and test
- 2: **for** epoch  $t = 1$  to  $T$  **do**
- 3:     **for** each batch  $\{(x_b, y_b)\}_{b=1:B}$  in train set **do**
- 4:         **for**  $m = 1$  to  $M$  **do** ▷ Forward passes with dropout
- 5:              $\hat{y}_{b=1:B}^{(m)} = model(x_{b=1:B}^{(m)})$  ▷  $\hat{y}_{b=1:B}^{(m)}$  is  $[B \times C]$  softmax outputs
- 6:         **end for**
- 7:         Estimate PEs for  $\{x_{b=1:B}^{(m=1:M)}\}$  using equation 6
- 8:         Compute Loss on  $\{(x_b^{(m)}, y_b^{(m)}, \hat{y}_b^{(m)})\}_{b=1:B}^{(m=1:M)}$  using equation 11
- 9:         Update weights by gradient descent
- 10:     **end for**
- 11:     Evaluate the model on test set
- 12: **end for**

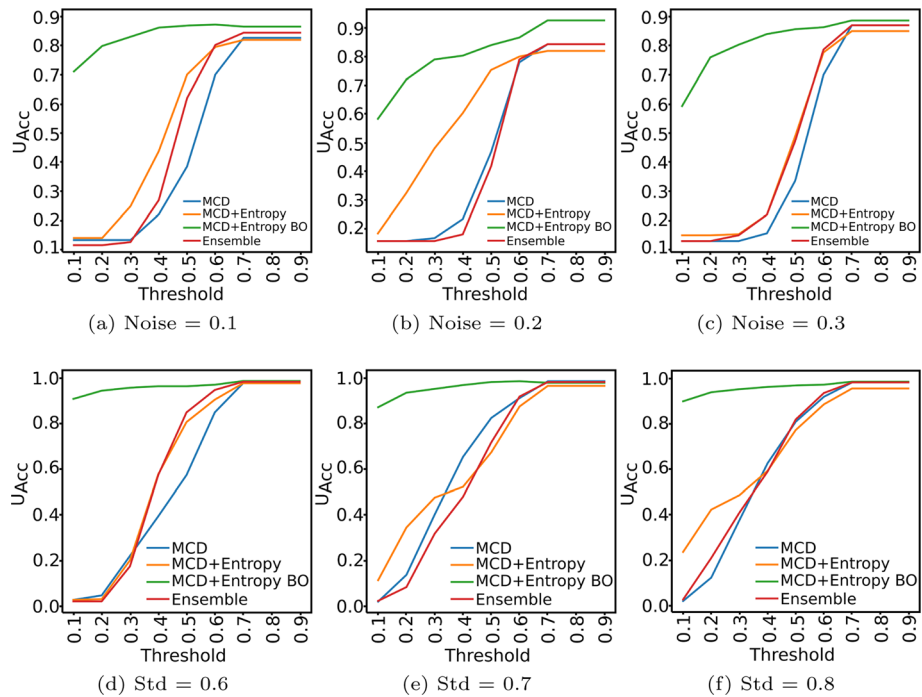
To further improve the performance of the method in algorithm 3, it is necessary to find appropriate values for dropout probabilities which can be done using Bayesian Optimization (BO). In general, hyper-parameter tuning can be done by either random search, grid search or BO. The grid search is a brute force method which is very time consuming but achieves good results. As the name implies, random search explores the space of hyper-parameters completely randomly, which may come up with reasonable values but not necessarily optimal ones. Contrary to random search and grid search, the BO takes a wiser approach by controlling the search direction based on obtained

observations. In the BO approach, an initial probability distribution is assumed in the space of hyper-parameter values. A sample is drawn from the aforementioned distribution and used as hyper-parameter values during the training. The BO distribution is then updated based on observed training loss. The update is done in an attempt to reduce the loss during the next training process. By repeating this process multiple times, the distribution in the hyper-parameter space keeps getting better and better.

**Fig. 3** UA metric for four algorithms on MNIST HAM-10000 dataset. UA is calculated for different uncertainty thresholds for different folds



**Fig. 4** UA metric for different algorithms and different thresholds for Two moons and Blobs dataset. The upper row denotes to two moons dataset and the below one is for Blobs dataset for different noise levels



**Table 2** UA and ECE of different Bayesian models averaged over six folds

UQ Method	UA %	ECE
MCD	70.42 ± 0.37	6.97 ± 0.32
MCD+Entropy	81.93 ± 0.28	2.48 ± 0.42
MCD+Entropy BO	83.03 ± 0.25	1.93 ± 0.3
Ensemble	72.45 ± 0.43	6.61 ± 0.5

### 4 Simulation and results

In this section, the proposed method is compared with three Bayesian architectures in terms of epistemic uncertainty quantification. To this end, Moons and Blobs synthetic datasets from Scikit Learn library and MNIST HAM-10000 (real) dataset related to skin cancer have been used in the experiments. HAM10000 (Human Against Machine with 10000 training images) dataset is available at Kaggle [24] and includes 10015 dermoscopy images of cancer and non-cancerous cases in different age ranges.

**Table 3** The qualitative comparison of four Bayesian models:  $\mu_1$  and  $\mu_2$  are the estimated centers of correct and incorrect distributions and *Dist* is the distance between the two centers

UQ Method	$\mu_1$	$\mu_2$	Dist	Acc %
MCD	0.33 ± 0.22	0.59 ± 0.18	0.25 ± 0.22	84.95 ± 0.23
MCD+Entropy	0.18 ± 0.21	0.48 ± 0.22	0.29 ± 0.23	85.39 ± 0.275
MCD+Entropy BO	0.16 ± 0.20	0.47 ± 0.21	0.30 ± 0.22	85.65 ± 0.18
Ensemble	0.31 ± 0.21	0.58 ± 0.21	0.27 ± 0.23	85.47 ± 0.28

For training and evaluation of the four Bayesian approaches, six-fold cross validation has been used. Given that deep models are data hungry and medical data are usually limited, using pre-trained models are advantageous. In our experiments, a DenseNet 121 pre-trained on ImageNet dataset was used for feature extraction. As preprocessing step, the images were resized to 224 × 224 and standardized (normalized). Relying on DenseNet121, 50176 convolutional features were extracted for each image which were fed to two dense layers. The activation functions were set to Relu. Optimization was done using Adam [25] algorithm. The experiments were run on Google Colab with its default settings (GPU: Tesla K80, 12GB GDDR5 VRAM).

To generate our ensemble model, number of neurons in the first and second dense layers were randomly selected from the sets {256, 257, ..., 512} and {32, 33, ..., 64}, respectively. The parameters of the two dense layers for each network in the ensemble were initialized randomly. The dropout probability was set to 0.25 in order to prevent overfitting during training. The hyperparameters of MCD

and MCD+entropy i.e. P1, P2, L1, and L2 were set to 0.25, 0.25, 128, and 64, respectively. It should be noted that P and L stand for dropout probability and number of neurons in each dense layer, respectively.

In our algorithm, the parameters of the model are optimized, similar to MCD+entropy algorithm. Additionally, the hyperparameters are optimized using BO in each training epoch. The resulting hyperparameters are reported in Table 1 for each dataset.

The estimated UA for different noise thresholds applied to the two synthetic datasets (Two Moons and Blobs) is available in Fig. 3. The top row is associated with the Two Moons dataset and the bottom row is related to the Blobs dataset with different values of standard deviation. Large standard deviation values mean that the two classes are heavily overlapped. The results for synthetic datasets reveal the superiority of our algorithm (MCD+entropy BO) over the other evaluated methods.

Since the real datasets are usually imbalanced, the performance of different architectures may vary for different subsets of data (splitting of data to test and train may affect). To tackle this issue, we use six fold cross validation to measure methods performance. The UA for all four algorithms has been depicted in Fig. 4 revealing that our algorithm outperforms its rivals by achieving better UA for different thresholds as we compare the results with others. Obviously, our algorithm (using optimized hyperparameters) improves the MCD+entropy algorithm for differentiating between correct and incorrect predictions (according to the UA's definition).

Table 2 reports expected values of UA and ECE computed over six folds (UA near one and ECE near zero is favorable). The lowest ECE belongs to MCD+entropy BO, which shows that this model is calibrated better than the other methods (ECE shows how much a specific model could produce calibrated predictions). Also, the UA of our method is higher than other methods revealing our model's ability to assign low PE to correct predictions and high PE to incorrect ones.

All four Bayesian methods have been compared in Table 3. Similar to Fig. 1, the expected correct and incorrect predictions ( $\mu_1$  and  $\mu_2$ ) and their distance (*Dist*) have been computed. Large *Dist* values for a model show its ability to distinguish between correct and incorrect distributions. As can be seen in Table 3, our proposed method has achieved higher *Dist* values demonstrating its superiority over the other three methods.

## 5 Conclusion

In this paper, we improved the uncertainty quantification performance of MCD method by proposing a novel approach for automatic optimization of deep model hyper-

parameters. The proposed approach was compared against three other Bayesian algorithms, namely, the MCD, the Ensemble Bayesian networks, and the MCD+entropy method for quantifying uncertainty associated with the skin cancer dataset. Experiments on the MNIST HAM10000 dataset showed that MCD+entropy BO method outperforms its rivals by offering reliable uncertainty without sacrificing the classification accuracy achieving  $85.65 \pm 0.18$  prediction accuracy,  $83.03 \pm 0.25$  uncertainty accuracy, and  $1.93 \pm 0.3$  expected calibration error outperforming vanilla MCD and MCD with a loss enhanced based on predicted entropy. Uncertainty quantification is of paramount importance for making trustworthy decisions under uncertainty in critical tasks of biomedical engineering. For the future works, we will use different heuristic optimization algorithms such as Whale and Grey Wolf to find out whether they can improve our proposed method.

**Acknowledgements** This work was funded by the UNSW Scientia Program Fellowship to HAR.

**Author contributions** HAR, AM, and AS designed the study; AS, HA, and ZB wrote the paper. The manuscript was edited by HAR, AS, KJ, MS, XW, IR, RA, and AM. AS, HA, RA, KJ carried out all the analyses. AS, HA and RA generated all figures and/or all tables. All authors have read and approved the final version of the paper.

**Funding** Open Access funding enabled and organized by CAUL and its Member Institutions.

**Data availability** The synthetic datasets (Two moons and Blobs) analysed during the current study are from Scikit Learn library and the skin cancer dataset (MNIST HAM-10000) is available at Kaggle [24].

## Declarations

**Conflict of interest** The authors declare no competing financial and non-financial interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Jerant AF, Johnson JT, Sheridan CD, Caffrey TJ (2000) Early detection and treatment of skin cancer. *Am Fam Physician* 62(2):357–368



2. Khaledyan D, Tajally A, Sarkhosh A, Shamsi A, Asgharnezhad H, Khosravi A, Nahavandi S (2021) Confidence aware neural networks for skin cancer detection. arXiv preprint [arXiv:2107.09118](https://arxiv.org/abs/2107.09118)
3. Rehman A, Naz S, Razzak I (2021) Leveraging big data analytics in healthcare enhancement: trends, challenges and opportunities. *Multimed Syst* 1–33:1339–1371
4. Razzak MI, Imran M, Xu G (2020) Big data analytics for preventive medicine. *Neural Comput Appl* 32(9):4417–4451
5. Shamsi A, Asgharnezhad H, Jokandan SS, Khosravi A, Kebria PM, Nahavandi D, Nahavandi S, Srinivasan D (2021) An uncertainty-aware transfer learning-based framework for covid-19 diagnosis. *IEEE Trans on Neural Netw and Learning Syst* 32(4):1408–1417
6. Nasab RZ, Ghamsari MRE, Argha A, Macphillamy C, Beheshti A, Alizadehsani R, Lovell NH, Alinejad-Rokny H (2022) Deep learning in spatially resolved transcriptomics: A comprehensive technical view. arXiv preprint [arXiv:2210.04453](https://arxiv.org/abs/2210.04453)
7. Hagenmüller S, Maron RC, Hekler A, Utikal JS, Barata C, Barnhill RL, Beltraminelli H, Berking C, Betz-Stablein B, Blum A et al (2021) Skin cancer classification via convolutional neural networks: systematic review of studies involving human experts. *Eur J Cancer* 156:202–216
8. Marchetti MA, Codella NC, Dusza SW, Gutman DA, Helba B, Kalloo A, Mishra N, Carrera C, Celebi ME, DeFazio JL et al (2018) Results of the 2016 international skin imaging collaboration international symposium on biomedical imaging challenge: Comparison of the accuracy of computer algorithms to dermatologists for the diagnosis of melanoma from dermoscopic images. *J Am Acad Dermatol* 78(2):270–277
9. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, Thrun S (2017) Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542(7639):115–118
10. Razzak I, Shoukat G, Naz S, Khan TM (2020) Skin lesion analysis toward accurate detection of melanoma using multistage fully connected residual network. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE
11. Razzak I, Naz S (2020) Unit-vise: deep shallow unit-vise residual neural networks with transition layer for expert level skin cancer classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*
12. Haenssle HA, Fink C, Schneiderbauer R, Toberer F, Buhl T, Blum A, Kalloo A, Hassen ABH, Thomas L, Enk A et al (2018) Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. *Ann Oncol* 29(8):1836–1842
13. Cheplygina V, de Bruijne M, Pluim JP (2019) Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Med Image Anal* 54:280–296
14. Abdar M, Pourpanah F, Hussain S, Rezaadegan D, Liu L, Ghavamzadeh M, Fieguth P, Cao X, Khosravi A, Acharya UR et al (2021) A review of uncertainty quantification in deep learning: techniques, applications and challenges. *Information Fusion* 76:243–297
15. Ye N, Zhu Z (2018) Functional bayesian neural networks for model uncertainty quantification
16. Shamsi A, Asgharnezhad H, Abdar M, Tajally A, Khosravi A, Nahavandi S, Leung H (2021) Improving mc-dropout uncertainty estimates with calibration error-based optimization. arXiv preprint [arXiv:2110.03260](https://arxiv.org/abs/2110.03260)
17. Asgharnezhad H, Shamsi A, Alizadehsani R, Khosravi A, Nahavandi S, Sani ZA, Srinivasan D, Islam SMS (2022) Objective evaluation of deep uncertainty predictions for covid-19 detection. *Sci Rep* 12(1):1–11
18. Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning, pp 1050–1059. PMLR
19. Seoh R (2020) Qualitative analysis of monte carlo dropout. arXiv preprint [arXiv:2007.01720](https://arxiv.org/abs/2007.01720)
20. Hasan M, Khosravi A, Hossain I, Rahman A, Nahavandi S (2022) Controlled dropout for uncertainty estimation. arXiv preprint [arXiv:2205.03109](https://arxiv.org/abs/2205.03109)
21. Dechesne C, Lassalle P, Lefèvre S (2021) Bayesian deep learning with monte carlo dropout for qualification of semantic segmentation. In: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, pp 2536–2539. IEEE
22. Ovadia Y, Fertig E, Ren J, Nado Z, Sculley D, Nowozin S, Dillon J, Lakshminarayanan B, Snoek J (2019) Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* 32
23. Lakshminarayanan B, Pritzel A, Blundell C (2017) Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems* 30
24. Skin Cancer MNIST: HAM10000. <https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000>
25. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

Afshar Shamsi<sup>1,6</sup> · Hamzeh Asgharnezhad<sup>5</sup> · Ziba Bouchani<sup>2</sup> · Khadijeh Jahanian<sup>3</sup> · Morteza Saberi<sup>3</sup> · Xianzhi Wang<sup>3</sup> · Imran Razzak<sup>4</sup> · Roohallah Alizadehsani<sup>5</sup> · Arash Mohammadi<sup>6</sup> · Hamid Alinejad-Rokny<sup>1,7,8</sup>

✉ Afshar Shamsi  
afshar.shamsi@encs.concordia.ca

✉ Hamid Alinejad-Rokny  
h.alinejad@unsw.edu.au

Hamzeh Asgharnezhad  
h.asgharnezhad@deakin.edu.au

<sup>1</sup> BioMedical Machine Learning Lab (BML), The Graduate School of Biomedical Engineering, UNSW Sydney, Sydney, Australia

<sup>2</sup> Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

<sup>3</sup> Faculty of Engineering and IT, University of Technology Sydney, Sydney, Australia

- <sup>4</sup> School of Computer Science and Engineering, The University of New South Wales (UNSW), Sydney, Australia
- <sup>5</sup> Intelligent for Systems Research and Innovation (IISRI), Deakin University, Waurn Ponds, Victoria 3216, Australia
- <sup>6</sup> Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada

- <sup>7</sup> UNSW Data Science Hub, The University of New South Wales (UNSW SYDNEY), Sydney, Australia
- <sup>8</sup> Health Data Analytics Program, Centre for Applied Artificial Intelligence, Macquarie University, Sydney, Australia

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)