Applied Soft Computing Journal 96 (2020) 106597

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Uncertainty quantification for plant disease detection using Bayesian deep learning

S. Hernández [a,b,*], Juan L. López [b]

[a] Laboratorio de Procesamiento de Información Geoespacial, Universidad Católica del Maule, Chile
[b] Centro de Innovación en Ingeniería Aplicada, Universidad Católica del Maule, Chile

ARTICLE INFO

Article history:
Received 2 January 2020
Received in revised form 26 May 2020
Accepted 29 July 2020
Available online 7 August 2020

Keywords:
Bayesian deep learning
Plant disease detection
Deep learning

ABSTRACT

Climate change is having an enormous impact on crop production in Latin America and the Caribbean. This problem not only concerns the volume of crop production but also the quality and safety of the food industry. Several research studies have proposed deep learning for plant disease detection. However, there is little information about the confidence of the prediction on unseen samples. Therefore, uncertainty in models of plant disease detection is required for effective crop management. In particular, uncertainty arising from sample selection bias makes it difficult to scale automatic plant disease detection systems to production. In this paper, we develop a probabilistic programming approach for plant disease detection using state-of-the-art Bayesian deep learning techniques and the uncertainty as a misclassification measurement. The results show that Bayesian inference achieves classification performance that is comparable to the standard optimization procedures for fine-tuning deep learning models. At the same time, the proposed method approximates the posterior density for the plant disease detection problem and quantify the uncertainty of the predictions for out-of-sample instances.

1. Introduction

Machine learning has been increasing its interest in areas such as bio-computing, computer vision, internet traffic, among others. In these and other areas the handling of large volumes of data is required. These facts lead to the development of increasingly complex and expensive computational models that solve specific tasks in certain application problems. Deep learning is a subset of the machine learning algorithms, that is especially well suited to analyze signals and images [1]. While traditional machine learning algorithms learn from examples that have been carefully crafted by human experts to represent a specific problem domain, deep learning algorithms can learn those representations from raw data. Image analysis and computer vision systems pose several external factors (such as illumination variations and complex background removal, among others) that make difficult to design human crafted representations to be used by a supervised machine learning algorithm [2]. In particular, Convolutional Neural Networks (CNN) are deep learning algorithms that overcome the cumbersome process related to manually creating features from images [3]. Deep learning architectures stack several processing layers (convolutions) that learn representations from the raw input data. However, as the model becomes more complex, the number of labeled examples that are required for effective learning is also increased.

Not surprisingly, modern computer vision tools for automatic crop disease diagnosis can be developed using deep learning techniques. These methods allow building end-to-end systems that digest leaf images and return the probability of the plant having any specific disease [4,5]. To successfully implement a deep learning-based disease diagnostic system, a large number of training examples with infected and healthy plants are required. Then, the performance of these methods is evaluated using standard cross-validation techniques (e.g. splitting the available examples into training and testing sets) and metrics such as classification accuracy, precision, and recall [6]. Using this methodology, plant disease detection using deep learning techniques has already shown outstanding performance (classification accuracy over 99%) [7]. This success can be attributed to the public availability of image databases with distinct combinations of sick/healthy plants. Notwithstanding the successful results, the reported accuracy is based on single estimates on the highest probability of an image belonging to a particular class. Therefore, an over-confident diagnostic may hinder the development of deep learning-based systems under realistic scenarios [8].

Most inference algorithms for deep learning are based on stochastic optimization techniques such as Stochastic Gradient Descent (SGD) [9]. These algorithms perform iterative updates

* Corresponding author at: Laboratorio de Procesamiento de Información Geoespacial, Universidad Católica del Maule, Chile.
E-mail address: shernandez@ucm.cl (S. Hernández).

https://doi.org/10.1016/j.asoc.2020.106597
1568-4946/

on the model parameters using a noisy estimate of the gradient of the objective function. In this setting, the main goal of the inference procedure is to obtain a point estimate of the model parameters. In particular, most practitioners are concerned with achieving the best out-of-sample error under a limited computational budget [10]. One of the main issues when taking machine learning models to the real world is the sample selection bias. For the plant disease detection problem, the training samples are usually collected in a constrained environment that does not represent the true data distribution.

Instead of relying on the traditional deep learning estimates, probabilistic inference using Bayesian techniques could be used to achieve well-calibrated uncertainty estimates [11]. Inference in Bayesian machine learning is a rather different approach since the goal is to obtain posterior probabilities instead of point estimates [12]. Since most of the models used in machine learning have no closed-form expressions, several Markov Chain Monte Carlo (MCMC) methods such as the Metropolis–Hastings algorithms, Gibbs sampling and Hamiltonian Monte Carlo are commonly used for Bayesian inference. Given that modern deep learning architectures must deal with large volumes of data, an issue for Bayesian deep learning systems based on neural networks is slow processing and poor scaling on high dimensional parameter spaces. Moreover, over-fitting may be present when the number of parameters is not compatible with the size of the data set and generalization on new data cannot be properly captured with the posterior samples. Dropout is a regularization technique for Deep neural networks, used during training time to avoid overfitting, that have been re-purposed as a variational Bayesian approximation [13], during test time, dropout is not applied. Conversely, Monte Carlo Dropout (MC Dropout) became an alternative to MCMC, applied at both training and test time, and has been successfully used to estimate uncertainty for computer vision models [14].

The uncertainty quantification is a great task. It is known that the validation of available data, the testing of different algorithms and sensitivity in the models increase the confidence captured by the uncertainty in the results. Achieving the necessary confidence requires better statistical interpretation under missing or inaccurate input data, appropriate strategies and models, and expert judgment. In [15], the authors point out 4 opportunity approaches in the task of improving the quantification of uncertainty: (1) Absence of theory or hard laws, (2) Absence of causal models, (3) Sensitivity to imperfect data and (4) Computational expense. The application and development of these solutions requires their extension to the Deep learning field to solve or mitigate the problems previously exposed. While uncertainty quantification appears as a necessary step for risk-sensitive tasks, previous works on plant disease detection have been mostly focused on estimating out-of-training error. Transfer learning using different deep learning architectures has been proposed for this task but uncertainty quantification has been given less attention. Therefore, the contributions of this paper are as follows:

- Bayesian deep learning has been previously proposed for uncertainty quantification in medical imaging classification. The importance in uncertainty quantification in medical imaging classification it is linking with probabilistic interpretations, so that under an insufficient understanding of the output model and overfitting in the neural network, the point prediction could provide suboptimal results, spuriously confidence, misclassifications and a incorrect decision making [16]. However, this is the first attempt to capture the uncertainty of deep learning for plant disease detection.
- We show that fine tuning a deep learning model on a plant diseases dataset produces over-confident predictions when there is a data mismatch with unseen data.
- We compare two different approaches for uncertainty quantification using Bayesian deep learning for plant disease detection, Monte Carlo Dropout and stochastic gradient Langevin Dynamics.

1.1. Related works

A baseline for detecting misclassified and out-of-sample data was proposed in [17]. The method uses probabilities from softmax distributions and the unseen data are expected to have lower classification probabilities (high entropy) when compared to the training data. Also, a (deep) ensemble of neural networks has been proposed for the task of uncertainty evaluation [18]. The deep ensembles approach also enables to identify out-of-distribution examples, when individual members of the ensemble mostly disagree. However, high entropy posterior distributions could also indicate class-overlap due to data unbalance [19].

In [20], the authors evaluate the MC Dropout for estimating diabetic retinopathy from fundus images. The authors show that uncertainty estimates from deep neural networks are useful for ranking out-of-sample data by their prediction performance. Uncertainty quantification using Bayesian neural networks in the medical imaging domain has been also studied in [21]. Predictive uncertainty is decomposed into its aleatoric and epistemic components while posterior samples are obtained using the variational dropout approximation.

Scalable MCMC techniques that exploit data sub-sampling using stochastic gradients were analyzed in [22]. These algorithms can be used to explore posterior distributions that provide good generalization capabilities. However, these methods suffer from poor mixing and require careful tuning for convergence. For this reason in this work, we propose to evaluate uncertainty estimates for plant disease detection using MCMC and compare those estimates with the dropout variational approximation.

2. BayesIan deep learning

In this work, a supervised classification dataset $\mathbf{D} = \{\mathbf{d_1}, \ldots, \mathbf{d_N}\}$ is represented by a set of tuples that contain the image features and the labels $d = (\mathbf{x}, y)$. The activation function for the softmax layer is a vector of probabilities $[\phi_1(\theta), \ldots, \phi_k(\theta)]$ and each one of the elements written as:

$$\phi_i(\theta) = \frac{\exp(\mathbf{x}^T \mathbf{w_i} + b_i)}{\sum_{j=1}^{K} \exp(\mathbf{x}^T \mathbf{w_j} + b_j)} \tag{1}$$

where $\theta = \{(\mathbf{w}_1, \mathbf{b}_1), \ldots, (\mathbf{w}_K, \mathbf{b}_K)\}$, where $\mathbf{w}_i \in \mathbf{R}^D$, b_i is a scalar, D represents the dimensionality of the feature space $\mathbf{x}$ and $K = 38$ the number of classes.

Finally, the objective function corresponds to the negative cross-entropy function (*log-loss*) defined as:

$$\log p(\mathbf{D}|\theta) = -\frac{1}{|D|} \sum_{\mathbf{d} \in \mathbf{D}} \sum_{k=1}^{K} \mathbf{1}_{y==k} \log \phi_k(\theta) \tag{2}$$

2.1. Stochastic gradient descent

In general, machine learning methods require a large number of examples to generalize and the numerical algorithms for optimizing the empirical loss must compute the gradient from Eq. (2). Batch optimization algorithms would make use of the full dataset $\mathbf{D}$ to compute the gradient. Conversely, stochastic gradient descent (SGD) assumes that the gradient is an expectation that can be estimated from a small batch of data $\mathbf{B}$.

Bayesian inference is concerned with obtaining posterior distributions for the model parameters $\boldsymbol{\theta}$ of a deep learning model

in the form of $p(\boldsymbol{\theta}|\mathbf{D})$. In this context, SGD with a constant learning rate has been interpreted as a Markov chain that converges to a stationary distribution [23]. The constant SGD algorithm iteratively performs parameter updates as follows.

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} + \epsilon\left(\nabla_\theta \log p(\boldsymbol{\theta}) + \widetilde{\nabla}_\theta \log p(\mathbf{D}|\boldsymbol{\theta})\right) \tag{3}$$

where ϵ is a constant learning rate and $\widetilde{\nabla}_\theta \log p(\mathbf{D}|\boldsymbol{\theta}) = -\frac{N}{|B_i|}\sum_{j\in B_i}\nabla_\theta \log p(\mathbf{d_j}|\boldsymbol{\theta})$ denotes a stochastic gradient evaluated using mini-batch B_i.

2.2. Stochastic gradient MCMC

Standard Monte Carlo methods for Bayesian data analysis make use of the full training data **D**. Nevertheless, this operation is too complex for large-scale datasets ($0 \ll N$). Instead, stochastic gradient MCMC (SG-MCMC) algorithms compute the approximate posterior using smaller data batches **B** (mini-batches) that suit a limited computational budget.

In particular, the stochastic gradient Langevin Dynamics (SGLD) algorithm makes use of simulations from a dynamical system to generate proposals [24]. The state of the system (current position of the parameter θ) moves along a trajectory according to the Langevin dynamics, while a time-varying learning rate is used to ensure that the proposals will always be accepted as $\epsilon_i \to 0$.

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} + \frac{\epsilon_i}{2}\left(\nabla_\theta \log p(\boldsymbol{\theta}) + \widetilde{\nabla}_\theta \log p(\mathbf{D}|\boldsymbol{\theta})\right) + \xi_i \tag{4}$$

where $\xi_i \sim \mathcal{N}(0, \epsilon_i)$ is an isotropic Gaussian random variable and ϵ_i is a time-varying learning rate.

In order to make a prediction y^* at a new test input $\mathbf{x}^*$, we need calculate the predictive distribution using a finite set of samples $\{\theta_t\}_{t=1}^T$ such as:

$$p(y^*|x^*, \mathbf{D}) \approx \int p(y^*|x^*, \theta)P(\theta|\mathbf{D})d\theta \tag{5}$$

$$\approx \frac{1}{T}\sum_{t=1}^{T} p(y^*|x^*, \theta_t)P(\theta_t|\mathbf{D}) \tag{6}$$

2.3. MC dropout

More recently, it has been shown that CNN models trained with dropout regularization is equivalent to variational approximate Bayesian inference [13]. The key idea to obtain model uncertainty is to train a model with dropout regularization and then perform dropout (Bernoulli) sampling with parameter p at test time. The empirical predictive distribution can be estimated by Monte Carlo sampling (MC dropout) and because of its simplicity, it is a good candidate for evaluating posterior distributions.

$$z \sim \mathcal{B}(p) \tag{7}$$

$$\theta_d = \theta_{\text{MAP}} \odot z \tag{8}$$

$$p(y^*|x^*, \mathbf{D}, p) \approx \frac{1}{T}\sum_{t=1}^{T} p(y^*|x^*, \theta_d)P(\theta_d|\mathbf{D}, p) \tag{9}$$

where θ_{MAP} is the MAP estimate and θ_d is the parameter vector θ after applying the Bernoulli mask z with dropout rate p.

3. Materials and methods

Localized detection of pests and/or fungi is highly desired. Achieving an accurate detection of pests and/or fungal population depends on the tools used for this purpose. Following the idea, to achieve the location of pests it is necessary for the uniform sampling of the plantations and testing with the information registered in a catalog of pests, fungi and/or diseases that could affect the crop [25].

A diagnosis based only on the agronomist experience can lead to erroneous conclusions and treatments. Due to this problem, and taking advantage of the great progress in recent years in the field of machine learning, deep learning techniques have been introduced in agriculture. In the field of plant disease diagnosis, it has only begun to gain ground to a fairly limited degree [26]. The previous problem opens a tremendous opportunity toward the implementation of computer systems based on deep learning that automatically detects and diagnoses diseases in plants [27].

3.1. Data source

Bayesian techniques for deep learning were evaluated on a database consisting of images of plant leaves. The PlantVillage dataset has been publicly released to foster research in this area and to compare results among different techniques. The dataset contains 54,306 images with combinations of 14 crops and 26 diseases, leaving a total number of 38 classes. The images consist of 256×256 pixels in the RGB color space. Fig. 1 shows a subset of the dataset.

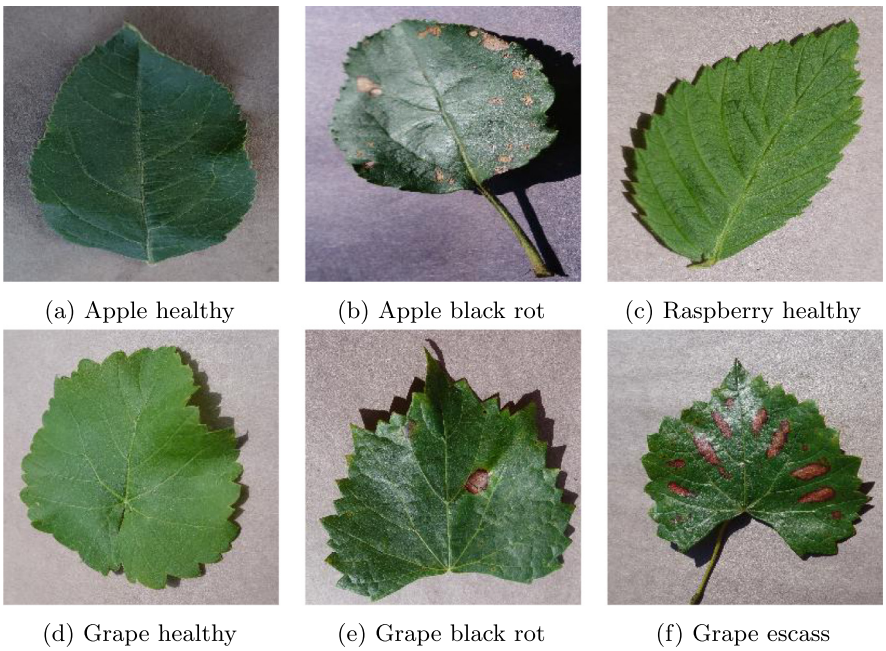
The full dataset is randomly split into training and testing sets, each one containing 80% and 20% of the data respectively. The choice of the dataset and the cross-validation split is based on [28] and [29].

3.2. Deep learning for plant disease detection

CNNs are specifically designed for analyzing multi-dimensional data such as images. Although the base model is inherited from the standard multi-layer neural network, a convolutional layer performs kernel operations over small areas of the image. The resulting representation is invariant to rotation and translation. When several convolutional layers are stacked, the model can be used to learn the features that best discriminate different object classes. This methodology has proven to be more robust than hand-crafted features that were previously used for plant disease detection [4].

Because of the sheer number of training examples required to fit deep learning models, the network is usually pre-trained from a large-scale dataset such as Imagenet (containing 1.2 million images and 1000 categories) and fine-tuning is used to train the final softmax layer to learn the 38 classes of the PlantVillage dataset. Too et al. [30] provide a complete description of the different deep learning architectures and the performance that can be achieved with this dataset. The authors compared state-of-the-art deep learning image classifiers such as VGG16, ResNet (with 50, 101 and 152 layers), Inception V4 and DenseNet. In particular, the VGG16 model stacks several convolutional layers and performs pooling operations on top [31]. This operation is repeated a number of times until a fully connected layer performs a softmax operation that compares the network output **x** and the training label y. An schematic overview of the proposed approach can be seen in Fig. 2.

To prevent the final layers to over-fit, different regularization schemes can be used. In particular, adding an L_2 norm term to the objective function and performing dropout sampling have proven to be effective when training fully connected networks such as VGG16 [32].



(a) Apple healthy (b) Apple black rot (c) Raspberry healthy

(d) Grape healthy (e) Grape black rot (f) Grape escass

Fig. 1. Leaf images from healthy and infected plants from the PlantVillage dataset.

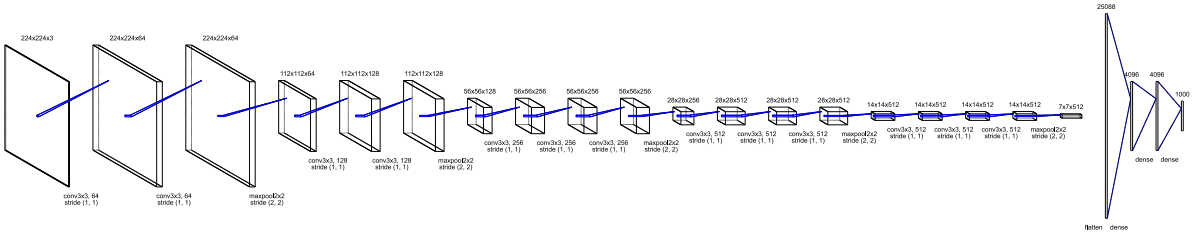


Fig. 2. Schematic representation of the VGG16 architecture. The model is first trained on the Imagenet dataset and the final layer can be replaced to recognize plant diseases.

4. Experiments

4.1. Bayeslan fine-tuning VGG16

A pre-trained VGG16 model using the Keras framework using weights obtained from training the Imagenet database is used to extract features from the train and test datasets. The final convolutional layer of the VGG16 model (also known as *conv_53*) consists of a $7 \times 7 \times 512$ tensor and is used to extract features. Each image is then represented by a vector of features $\mathbf{x} \in \mathbb{R}^{25088}$.

First, we evaluate the performance of using SGD for training a softmax layer with a normal prior (which is equivalent to adding an L_2 regularization term). Weights and biases are initialized to zero in all experiments. A total of 100 epochs is run with a constant learning rate $\epsilon = 1e^{-5}$ and batch size $|B| = 100$. The prior on $\boldsymbol{\theta} \sim \mathcal{N}(0, \frac{1}{\alpha} I)$ is set to a multivariate Gaussian with diagonal covariance and $\alpha = 1e^{-2}$. MC dropout with parameter $p = 0.5$ is also evaluated.

The log-loss metric is frequently used to describe how well the parameters of the model fits a set of observations. SGD with a constant learning rate achieves convergence after 60 epochs. However, since SGD is equivalent to maximum likelihood training it might not capture the uncertainty [33]. On the other hand, SGLD and MC dropout achieve a less confident estimate. Fig. 3 shows the log-loss function (negative cross-entropy) of the fully connected layer of the VGG16 model.

The average log-loss metric is an indicator of the mean predictive accuracy, but the posterior variance can be different depending on the inference algorithm. Table 1 shows the mean and standard deviation of the predictive accuracy for all of the methods.

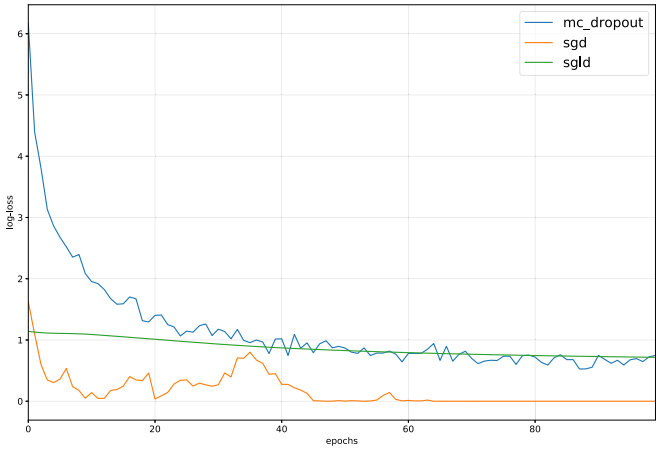


Fig. 3. Comparing different regularization schemes for fine-tuning the VGG16 model.

4.2. Uncertainty quantification

Uncertainty in predictions has source both in the generation of data (noise) and/or the ability to get a representative model

Table 1
Performance of fine-tuning VGG16 using Bayesian learning schemes.

Method	Accuracy	Recall	$F1$ score	Support
SGD	0.96 ± 0.00	0.96 ± 0.00	0.96 ± 0.00	10848
MC dropout	0.94 ± 0.02	0.94 ± 0.02	0.94 ± 0.02	10848
SGLD	0.89 ± 0.05	0.89 ± 0.05	0.88 ± 0.06	10848

of them (uncertainty in the model) [34]. The uncertainty caused by both sources can be reduced by increasing the number of data available while bias is minimized. Although the development of machine learning tools has meant a great advance, the uncertainty present on models leaves a large gap that should be explored. Quantifying uncertainty could provide a background of information when making decisions [35] and it is a great help in those cases where our models do not generate the expected results. There are different approaches to quantify uncertainty and the entropy by variational dropout is one of them [36].

Entropy by variational dropout has been previously explored by Raczkowski et al. [37] and Kwon et al. [21]. This technique focuses on continuously apply dropout regularization in each phase of training, so that by discarding different neurons (units) in the network, chosen randomly, we would be in the presence of a Bayesian sampling from a variational distribution of models.

The Shannon entropy H of the predictive distribution [13] can be defined by:

$$H[p(y^*|\mathbf{x}^*, \mathbf{D})] = -\sum_{y^* \in 1..M} p(y^*|\mathbf{x}^*, \mathbf{D}) \log p(y^*|\mathbf{x}^*, \mathbf{D}) \tag{10}$$

where $p(y^*|\mathbf{x}^*, \mathbf{D})$ can be approximated by the variational predictive distribution given by

$$\kappa(y^*|\mathbf{x}^*) \approx \int p(y^*|\mathbf{x}^*, \theta)\kappa(\theta)d\theta \approx \frac{1}{T}\sum_{t=1}^{T} p(y^*|\mathbf{x}^*, \theta_t) \tag{11}$$

when T goes to infinity.

During the training process, $T = 100$ samples were used for each one of the methods. The dataset was grouped by class; Apple, Cherry, Corn, Grape, Peach Pepper, Potato, Strawberry and Tomato. Healthy and non-healthy species are available for each grouped dataset. As mentioned in Section 4.2, quantifying uncertainty could improve decision making. Therefore, the aim is to understand the outcomes from MC dropout and SGLD methods as well as the magnitude and range where the predictions are valid.

The results of Table 2 show that relative maximum fluctuation $\Phi H^{(\alpha)}_{\max}$ from the local mean is even higher than 5% and less than 23%, the best performing is the Tomato class with 5% and the worse are the Grape and Raspberry classes with 23%.

To visualize easier this last point, for all α-class predictions the Shannon entropy H is computed MC dropout($H^{(\alpha)}_{MCd}$) and SGLD ($H^{(\alpha)}_{SGLD}$) methods for all α-class predictions.

Fig. 4 shows H computed to four representative class by MC dropout (left panel) and SGLD (right panel) methods. Clearly H_{MCd} (left panel Fig. 4) shows small fluctuation around mean value over all processed samples. This behavior shows the stochastic nature of the uncertainties due to the random selection of the samples and the independence of each training stage. In contrast with H_{MCd}, for all α-class, the behavior of H_{SGLD} shows that uncertainty reaches a maximum value before the first 20 processed samples then decays with a number of them (right panel Fig. 4). Based on these results and the differences between H_{MCd} and H_{SGLD} behaviors, it is tempting to advance the hypothesis that SGLD exhibits a long-range memory and class predictions improve only when a large number of samples are processed, H_{SGLD} value approximates to H_{MCd} and MC dropout outperforms SGLD.

Based on performance and stability shown in uncertainty for class prediction by MC dropout, some quantities have been computed to quantify somehow the uncertainty behavior for all class and processed sample: the local mean value $\langle H\rangle_{MCd}$, the corresponding standard deviation $\sigma(H_{MCd})$, relative maximum fluctuation to α-class

$$\Phi H^{(\alpha)}_{\max} \equiv |\max\{H^{(\alpha)}_{MCd}\} - \langle H\rangle^{(\alpha)}_{MCd}|/\langle H\rangle^{(\alpha)}_{MCd} \tag{12}$$

and relative difference of $\langle H\rangle^{(\alpha)}_{MCd}$ respect to healthy class ($\alpha = h$)

$$\Delta H^{(h)}_{MCd} \equiv |\langle H\rangle^{(\alpha)}_{MCd} - \langle H\rangle^{(h)}_{MCd}|/\langle H\rangle^{(h)}_{MCd} = |\langle H\rangle^{(\alpha-h)}_{MCd}|/\langle H\rangle^{(h)}_{MCd} \tag{13}$$

To relative difference from healthy class, the less difference is around 0.07 to "Grape Leaf blight (Isariopsis Leaf spot)" while the higher is around 5.22 to "Grape Black rot" and special care should be taken in those cases where $\Delta H_{\text{healthy}} \approx 0$ is not contrasting factors to distinguishing between different diseases for each particular species. When all crop class are analyzed and compared between them, it is no longer possible to get significant differences and the predicted class can be an incorrect classification (see Table 3). The results shown in Table 3 summarizes the absolute difference of local mean values $|\langle H\rangle^{(\alpha_i-\alpha_j)}_{MCd}| = |\langle H\rangle^{(\alpha_i)}_{MCd} - \langle H\rangle^{(\alpha_j)}_{MCd}|$, the relative difference $\Delta H^{(\alpha_j)}_{MCd}$ and the significance level of the real class α_i respect to predicted class α_j for those cases in which is not significant difference between species, and were the predicted classes were misclassified. It is important to mention that in each summarized cases, where the predicted classes were misclassified, the relative difference $\Delta H^{(\alpha_i)}_{MCd}$ was equal to or less than 2%.

4.3. Domain adaption

Realistic implementations of deep learning based-plant disease detection systems must take into account some important factors. Barbedo et al. [8] identified covariate shift and image quality from field conditions among other factors to be considered. Several deep learning models have been proposed for the PlantVillage dataset. These models achieved outstanding performance in the out-of-training error when the same database is used for training and testing. Covariate shift occurs when a predictive model is trained with data that differs from field conditions. The authors in [28] shown that a model trained with the PlantVillage dataset (whose images were taken under controlled conditions), decreases its performance when tested with images from other sources. On the other hand, the amount of labeled data is still insufficient to capture the different signs and symptoms of crop diseases. Collecting more data under a wide variety of conditions could alleviate this problem, but data labeling is both expensive and time-consuming [38].

To assess the role of uncertainty quantification for generalization performance we use an image dataset of Northern Leaf Blight (NLB) infected maize leaves [39]. The full dataset consists of 1787 annotated images with 7669 lesions. The images were captured from maize plants in the field using a portable handheld camera and were manually labeled by two human experts. Fig. 5 shows northern leaf blight infected plants.

Domain adaption is a particular kind of covariate shift that occurs when the distribution of the inputs varies among the training and testing procedures. In the context of a plant disease recognition system, a classifier trained from a source domain is expected to deal with images captured in a broad variety of conditions. In particular, images from the PlantVillage dataset were obtained in a controlled environment where a single leaf is placed on a plain background and lightning conditions were carefully taken into account. Although the NLB and the PlantVillage

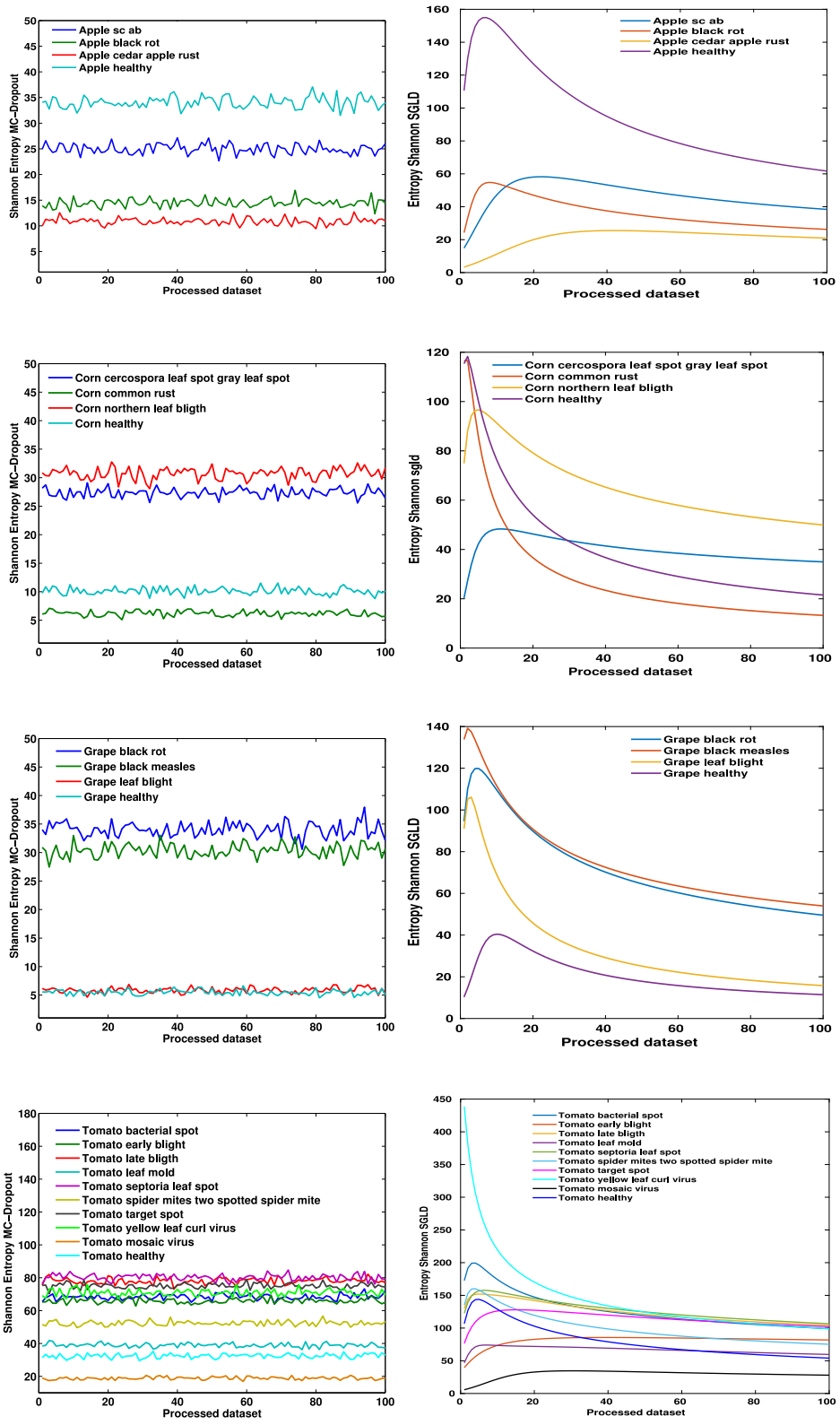


Fig. 4. Entropy Leaf images from healthy and infected plants from the PlantVillage dataset by MC dropout and SGLD.

datasets contain overlapped classes (healthy and non-healthy maize plants), the images in the NLB dataset were captured in uncontrolled conditions and with no focus on a particular plant organ [40]. To understand the generalization performance, all models that were previously trained with the PlantVillage dataset are used to classify images from the NLB dataset. The performance of the out-of-sample classification problem can be seen in Table 4.

Table 4 shows that SGD yields worse performance for the out-of-sample classification problem. MC dropout with dropout rate

Table 2
Mean and standard variation for H in each class as well as the corresponding values of $\Phi H^{(\alpha)}_{\max}$ and $\Delta H^{(h)}_{MCd}$.

Name	$\langle H\rangle^{(\alpha)}_{MCd}$	$\sigma(H_{MCd})$	$\Phi H^{(\alpha)}_{\max}$	$\Delta H^{(h)}_{MCd}$
Apple___healthy	**34.06**	1.11	0.09	–
Apple___Apple_scab	24.99	0.95	0.09	0.26
Apple___Black_rot	14.46	0.79	0.17	0.57
Apple___Cedar_apple_rust	10.86	0.64	0.17	0.68
Corn_(maize)___healthy	**10.11**	0.61	0.14	–
Corn_(maize)___Cercospora_leaf_spot Gray_leaf	27.37	0.77	0.06	1.70
Corn_(maize)___Common_rust	6.19	0.47	0.15	0.39
Corn_(maize)___Northern_Leaf_Blight	30.74	0.97	0.07	2.04
Grape___healthy	**5.47**	0.45	0.21	–
Grape___Black_rot	34.00	1.26	0.12	5.22
Grape___Esca_(Black_Measles)	30.35	1.16	0.09	4.55
Grape___Leaf_blight_(Isariopsis_Leaf_Spot)	5.87	0.48	0.17	0.07
Potato___healthy	**9.48**	0.50	0.11	–
Potato___Early_blight	20.07	0.98	0.15	1.12
Potato___Late_blight	45.02	1.25	0.07	3.74
Tomato___healthy	**32.42**	1.32	0.09	–
Tomato___Bacterial_spot	68.35	1.64	0.07	1.11
Tomato___Early_blight	65.92	1.44	0.05	1.03
Tomato___Late_blight	78.02	1.73	0.05	1.41
Tomato___Leaf_Mold	38.94	1.22	0.07	0.20
Tomato___Septoria_leaf_spot	80.29	1.90	0.05	0.24
Tomato___Spider_mites Two-spotted_spider_mite	52.33	1.41	0.08	1.48
Tomato___Target_Spot	75.07	1.57	0.05	1.32
Tomato___Tomato_Yellow_Leaf_Curl_Virus	70.95	1.96	0.07	1.19
Tomato___Tomato_mosaic_virus	18.86	0.85	0.09	0.45
Cherry_(including_sour)___healthy	**12.12**	0.78	0.16	–
Cherry_(including_sour)___Powdery_mildew	17.50	0.83	0.12	0.44
Peach___healthy	**13.05**	0.72	0.13	–
Peach___Bacterial_spot	29.99	1.37	0.11	1.30
Pepper,_bell___healthy	**28.09**	1.08	0.13	–
Pepper,_bell___Bacterial_spot	37.39	1.17	0.09	0.33
Strawberry___healthy	**8.58**	0.54	0.14	–
Strawberry___Leaf_scorch	20.23	0.93	0.11	1.36
Blueberry___healthy	**15.94**	0.85	0.17	–
Raspberry___healthy	**4.99**	0.39	0.23	–
Soybean___healthy	**29.78**	1.30	0.10	–
Orange___Haunglongbing_(Citrus_greening)	18.26	0.96	0.14	–
Squash___Powdery_mildew	18.05	0.81	0.17	–

Table 3
Cases in which there is not significant difference between species and the predicted class was incorrect.

(α_i) class	(α_j) class	$\lvert\langle H\rangle^{(\alpha_i-\alpha_j)}_{MCd}\rvert$	$\Delta H^{(\alpha_j)}_{MCd}$	$Sig.$
Grape Black rot	Apple healthy	0.06	0.02	1.00
Grape Black Measles	Corn Northern Leaf Blight	0.39	0.01	0.83
Peach Bacterial spot	Grape Black Measles	0.36	0.01	0.99
Soybean healthy	Grape Black Measles	0.57	0.02	0.28
Squash powdery mildew	Orange Haunblongbing	0.21	0.01	1.00
Soybean healthy	Peach Bacterial spot	0.22	0.01	1.00
Potato Early blight	Strawberry Leaf scorch	0.16	0.01	1.00

Fig. 5. Leaf images from northern leaf blight infected maize plants.

$p = 0.5$ improves the mean accuracy and SGLD improves the recall metric. Now, we compare the accuracy results of the different methods using a point estimate. Fig. 6 shows the confusion matrices for each one of the methods using the posterior median as a point estimate.

The posterior mean probabilities are used to classify images from the NLB dataset into one of the PlantVillage classes. For

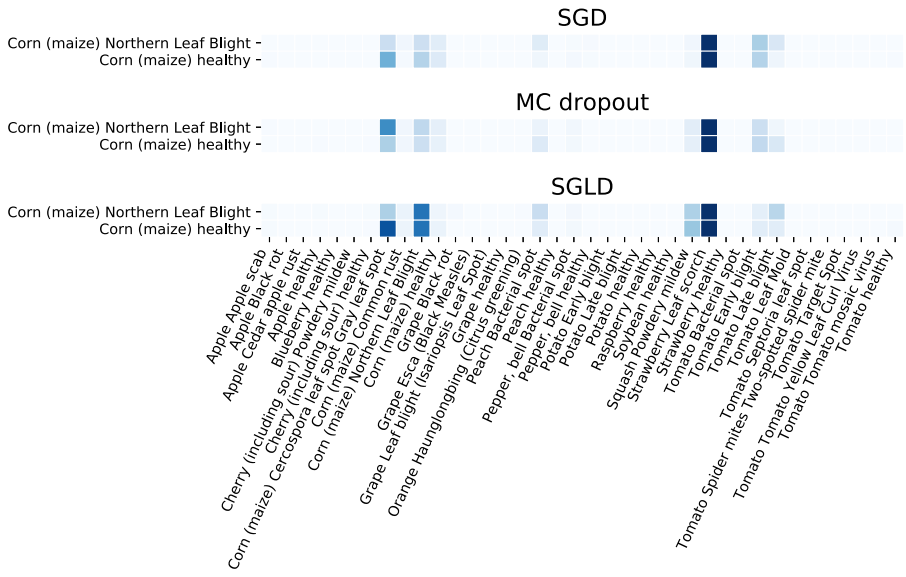


Fig. 6. Confusion matrices for the out-of-sample classification problem.

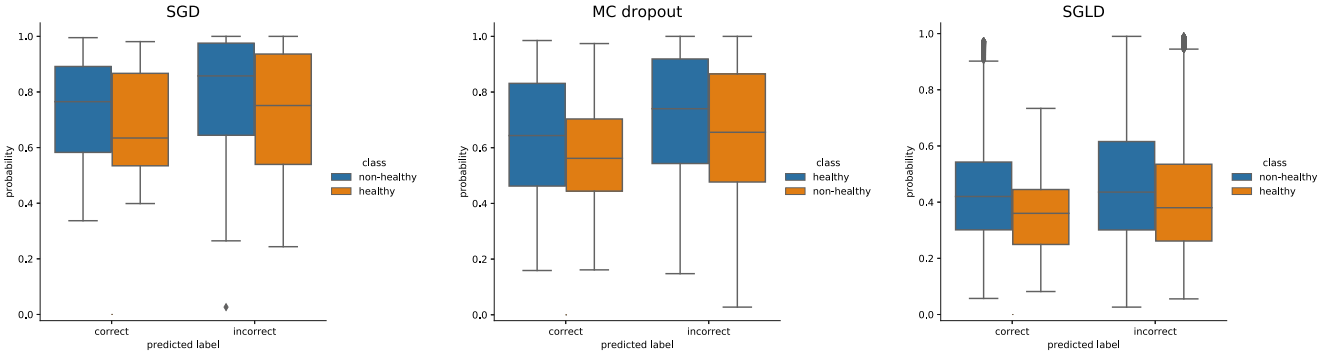


Fig. 7. Maximum softmax probabilities for SGD, MC dropout and SGLD.

Table 4
Performance of out-of-sample classification.

Method	Accuracy	Recall	$F1$ score	Support
SGD	0.49 ± 0.00	0.05 ± 0.00	0.09 ± 0.00	1787
MC dropout	0.55 ± 0.02	0.05 ± 0.00	0.1 ± 0.006	1787
SGLD	0.54 ± 0.01	0.1 ± 0.02	0.15 ± 0.02	1787

Table 5
Mean softmax probabilities for SGD, MC dropout and SGLD for correctly and incorrectly classified out-of-sample examples.

	Healthy		Non-healthy	
Method	Correct	Incorrect	Correct	Incorrect
SGD	0.68 ± 0.19	0.73 ± 0.21	0.72 ± 0.19	0.79 ± 0.20
MC dropout	0.64 ± 0.21	0.72 ± 0.21	0.58 ± 0.17	0.66 ± 0.22
SGLD	0.36 ± 0.14	0.41 ± 0.19	0.43 ± 0.17	0.47 ± 0.21

the healthy/non-healthy classification task we use the maximum softmax probability and assign the predicted label to each example. Samples are separated into correctly and incorrectly classified. Fig. 7 shows the maximum probabilities used for the classifying out-of-sample data.

SGD and MC dropout tend to produce overconfident incorrect predictions whose maximum probabilities tend to be larger than their correct counterparts. In the other hand, SGLD produces less confident outcomes for both the correctly and incorrectly classified samples. Table 5 shows the mean of the softmax probabilities.

5. Conclusions

In conclusion, quantifying uncertainty provide a valuable background of information on decisions making. For MCd and SGLD methods, the uncertainty predictions for each species and training stage was quantified by the Shannon entropy H of the variational predictive distribution. The best predictions are reached when MC dropout methods are used, then Shannon entropy can be used to quantify uncertainty. The uncertainty showed a slight fluctuation of stochastic nature concerning the mean value of all α-class were the correct class prediction was made. In contrast, the SGLD method exhibits long-range memory for all of the α-class. H_{SGLD} reaches stability as the number of processed samples increases and H_{SGLD} has a value close to the abandonment of H_{MCd}. The relative maximum fluctuation $\Phi H^{(\alpha)}_{\max}$ from the local mean was around than 5% on the best performing and than 23% on the worst cases. Outside these regions not only is the agreement is worst, but also the results could led to a wrong assignment of a predicted class. For those cases where the predicted class was misclassified, the relative difference $\Delta H^{(\alpha_i)}_{MCd}$ was equal to or less than 2%. When the realistic implementations of deep learning based-plant disease detection were made, the covariate shift and image quality from field conditions among other factors, the performance decreases. On particular, SGD and MC dropout tend

to produce overconfident incorrect predictions whose maximum probabilities tend to be larger than their correct counterparts. In the other hand, SGLD produces less confident outcomes for both the correctly and incorrectly classified samples.

CRediT authorship contribution statement

S. Hernández: Conceptualization, Methodology, Software, Writing - review & editing. **Juan L. López:** Writing, Visualization, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep Learning, MIT Press, 2016.
[2] Yoshua Bengio, Aaron Courville, Pascal Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[3] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[4] Andreas Kamilaris, Francesc X. Prenafeta-Boldu, Deep learning in agriculture: A survey, Comput. Electron. Agric. 147 (2018) 70–90.
[5] Zahid Iqbal, Muhammad Attique Khan, Muhammad Sharif, Jamal Hussain Shah, Muhammad Habib ur Rehman, Kashif Javed, An automated detection and classification of citrus plant diseases using image processing techniques: A review, Comput. Electron. Agric. 153 (2018) 12–32.
[6] Liangxiu Han, Muhammad Salman Haleem, Moray Taylor, Automatic detection and severity assessment of crop diseases using image pattern recognition, in: Emerging Trends and Advanced Technologies for Computational Intelligence, Springer, 2016, pp. 283–300.
[7] Konstantinos P. Ferentinos, Deep learning models for plant disease detection and diagnosis, Comput. Electron. Agric. 145 (2018) 311–318.
[8] Jayme G.A. Barbedo, Factors influencing the use of deep learning for plant disease recognition, Biosyst. Eng. 172 (2018) 84–91.
[9] Herbert Robbins, Sutton Monro, A stochastic approximation method, Ann. Math. Stat. 22 (3) (1951) 400–407.
[10] Christopher J. Shallue, Jaehoon Lee, Joseph Antognini, Jascha Sohl-Dickstein, Roy Frostig, George E Dahl, Measuring the effects of data parallelism on neural network training, J. Mach. Learn. Res. 20 (112) (2019) 1–49.
[11] Andrew Gelman, Hal S. Stern, John B. Carlin, David B. Dunson, Aki Vehtari, Donald B. Rubin, Bayesian Data Analysis, third ed., Chapman and Hall/CRC, 2013.
[12] David J.C. MacKay, Information Theory, Inference & Learning Algorithms, Cambridge University Press, New York, NY, USA, 2002.
[13] Yarin Gal, Zoubin Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, 2016, pp. 1050–1059.
[14] Alex Kendall, Yarin Gal, What uncertainties do we need in Bayesian deep learning for computer vision? in: Advances in Neural Information Processing Systems, 2017, pp. 5574–5584.
[15] Edmon Begoli, Tanmoy Bhattacharya, Dimitri Kusnezov, The need for uncertainty quantification in machine-assisted medical decision making, Nature (2019).
[16] Jiawei Su, Danilo Vargas, Kouichi Sakurai, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Comput. PP (2017).
[17] Dan Hendrycks, Kevin Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, 2016, arXiv preprint arXiv:1610.02136.
[18] Balaji Lakshminarayanan, Alexander Pritzel, Charles Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: Advances in Neural Information Processing Systems, 2017, pp. 6402–6413.
[19] Andrey Malinin, Mark Gales, Predictive uncertainty estimation via prior networks, in: Advances in Neural Information Processing Systems, 2018, pp. 7047–7058.
[20] C. Leibig, V. Allken, M.S. Ayhan, P. Berens, S. Wahl, Leveraging uncertainty information from deep neural networks for disease detection, Sci. Rep. 7 (1) (2017) 17816.
[21] Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, Myunghee Cho Paik, Uncertainty quantification using Bayesian neural networks in classification: Application to biomedical image segmentation, Comput. Statist. Data Anal. 142 (2020) 106816.
[22] Yi-An Ma, Tianqi Chen, Emily B. Fox, A complete recipe for stochastic gradient MCMC, in: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15, MIT Press, Cambridge, MA, USA, 2015, pp. 2917–2925.
[23] Stephan Mandt, Matthew D. Hoffman, David M. Blei, Stochastic gradient descent as approximate Bayesian inference, J. Mach. Learn. Res. 18 (1) (2017) 4873–4907.
[24] Max Welling, Yee W. Teh, Bayesian learning via stochastic gradient Langevin dynamics, in: Proceedings of the 28th International Conference on Machine Learning, ICML-11, 2011, pp. 681–688.
[25] M.B. Sposito, L. Amorim, R.B. Bassanezi, A.B. Filho, B. Hau, Spatial pattern of black spot incidence within citrus trees related to disease severity and pathogen dispersal, Plant Pathol. 57 (2008) 103–108.
[26] S.P. Mohanty, D.P. Hughes, M. Salathé, Using deep learning for image-based plant disease detection, Front. Plant Sci. 7 (2016) 1419.
[27] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature Cell Biol. 521 (7553) (2015) 436–444.
[28] Sharada P. Mohanty, David P. Hughes, Marcel Salathe, Using deep learning for image-based plant disease detection, Front. Plant Sci. 7 (2016) 1419.
[29] Edna Chebet Too, Li Yujian, Sam Njuki, Liu Yingchun, A comparative study of fine-tuning deep learning models for plant disease identification, Comput. Electron. Agric. (2018).
[30] G. Wang, Y. Sun, J. Wang, Automatic image-based plant disease severity estimation using deep learning, Comput. Intell. Neurosci. 2017 (2017).
[31] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[33] Wesley J. Maddox, Pavel Izmailov, Timur Garipov, Dmitry P. Vetrov, Andrew Gordon Wilson, A simple baseline for Bayesian uncertainty in deep learning, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 13132–13143.
[34] David Draper, Assessment and propagation of model uncertainty, J. R. Stat. Soc. Ser. B Stat. Methodol. 57 (1) (1995) 45–97.
[35] Abbas Khosravi, Saeid Nahavandi, Douglas C. Creighton, Amir F. Atiya, Comprehensive review of neural network-based prediction intervals and new advances, IEEE Trans. Neural Netw. 22 (2011) 1341–1356.
[36] Yarin Gal, Zoubin Ghahramani, Bayesian convolutional neural networks with Bernoulli approximate variational inference, 2015, arXiv preprint arXiv:1506.02158.
[37] L. Raczkowski, M. Mozejko, J. Zambonelli, ARA: accurate, reliable and active histopathological image classification framework with Bayesian deep learning, Sci. Rep. 9 (2019) 14347.
[38] Konstantinos P. Ferentinos, Deep learning models for plant disease detection and diagnosis, Comput. Electron. Agric. 145 (2018) 311–318.
[39] Tyr Wiesner-Hanks, Ethan L. Stewart, Nicholas Kaczmar, Chad DeChant, Harvey Wu, Rebecca J. Nelson, Hod Lipson, Michael A. Gore, Image set for deep learning: field images of maize annotated with disease symptoms, BMC Res. Notes 11 (1) (2018) 440.
[40] Justine Boulent, Samuel Foucher, Jérôme Théau, Pierre-Luc St-Charles, Convolutional neural networks for the automatic identification of plant diseases, Front. Plant Sci. 10 (2019) 941.