applied sciences

MDPI

Article

Uncertainty Quantification for MLP-Mixer Using Bayesian Deep Learning

Abdullah A. Abdullah [1,*], Masoud M. Hassan [1] and Yaseen T. Mustafa [2,3]

[1] Computer Science Department, Faculty of Science, University of Zakho, Duhok 42002, Kurdistan Region, Iraq
[2] Computer Science Department, College of Science, Nawroz University, Duhok 42001, Kurdistan Region, Iraq
[3] Environmental Science Department, Faculty of Science, University of Zakho, Duhok 42002, Kurdistan Region, Iraq
* Correspondence: abdullah.abdullah@uoz.edu.krd

Abstract: Convolutional neural networks (CNNs) have become a popular choice for various image classification applications. However, the multi-layer perceptron mixer (MLP-Mixer) architecture has been proposed as a promising alternative, particularly for large datasets. Despite its advantages in handling large datasets and models, MLP-Mixer models have limitations when dealing with small datasets. This study aimed to quantify and evaluate the uncertainty associated with MLP-Mixer models for small datasets using Bayesian deep learning (BDL) methods to quantify uncertainty and compare the results to existing CNN models. In particular, we examined the use of variational inference and Monte Carlo dropout methods. The results indicated that BDL can improve the performance of MLP-Mixer models by 9.2 to 17.4% in term of accuracy across different mixer models. On the other hand, the results suggest that CNN models tend to have limited improvement or even decreased performance in some cases when using BDL. These findings suggest that BDL is a promising approach to improve the performance of MLP-Mixer models, especially for small datasets.

Keywords: uncertainty quantification; Bayesian deep learning; MLP-Mixer; variational inference (VI); MC-dropout

check for updates

Citation: Abdullah, A.A.; Hassan, M.M.; Mustafa, Y.T. Uncertainty Quantification for MLP-Mixer Using Bayesian Deep Learning. *Appl. Sci.* **2023**, *13*, 4547. https://doi.org/10.3390/app13074547

Academic Editor: Yu-Dong Zhang

Received: 11 March 2023
Revised: 27 March 2023
Accepted: 28 March 2023
Published: 3 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In recent years, convolutional neural networks (CNNs) have been adopted as the most widely used architecture for computer vision problems. This adaptation of CNNs in the field of image processing is mainly due to their effectiveness and efficiency [1]. These networks are composed of layers that utilize convolutional methods on an input to produce an output. Filters, which operate using a sliding window on the input, are utilized in the convolution process to generate a feature map [2]. CNNs often incorporate pooling layers, in conjunction with convolution layers, to reduce the dimensionality of the feature map and decrease computational demands. A tensor with one, two, or three dimensions can be used as input for the CNN, and the output produces similar shapes in most cases [3]. Various CNN architectures have been developed in the last decade, including ResNet [4], Inception [5], and DenseNet [2]. In the field of biomedical deep learning, CNNs have seen widespread adoption and have been applied to numerous applications, such as medical imaging.

Over time, researchers have developed several CNN models and architectures, such as ResNet [4], Inception [5], DenseNet [6], vision transformers [7], and Swin transformers [8], which have become increasingly complex. Despite the popularity of CNNs, some alternative architectures have emerged recently, such as the multi-layer perceptron mixer (MLP-Mixer), which was introduced in 2021. A recent study by Tolstikhin et al. [9] challenged the notion that more complex models and architectures lead to better performance. The authors introduced a novel architecture called MLP-Mixer, which is based on multi-layer perceptrons (MLPs) in conjunction with mixers. The MLP-Mixer model divides

images into patches, referred to as tokens, which then transform into simple MLPs using matrix multiplication. The use of two layers of mixers, token mixing, and channel mixing, enabled inter- and intra-interaction between patches. While MLP-Mixers have shown state-of-the-art performance and perform well on large datasets, they still lag behind CNN models on small datasets and have a larger number of parameters [9,10]. To address these issues, researchers have proposed various versions of the MLP-Mixer model, such as Sparse-MLP, S2-MLP, As-MLP, and Hire-MLP [10]. Despite these challenges, the key advantages of the MLP-Mixer are its simplicity, flexibility, and scalability, which make it an attractive alternative to CNNs for vision tasks.

In the field of deep learning for computer vision, the performance of various architectures, such as MLP-Mixers and CNNs, is typically evaluated using measurements such as sensitivity, specificity, accuracy, precision, recall, receiver operating characteristics (ROCs), area under the ROC curve (AUC), F1 score, and other similar metrics [11]. Although these metrics have been successful in differentiating between good and bad models, they do not provide any information about the confidence of a particular model. Measuring the confidence of a model is crucial, especially in domains such as medical diagnosis, where errors can have serious consequences. Thus, it is essential to estimate the uncertainty of both the model and the data. One method for uncertainty quantification is through the Bayesian deep learning (BDL) models [12], which treats the weights of the model as random variables. BDL incorporates prior information about the distribution of the weights and updates it with new data to estimate the uncertainty associated with the model's predictions.

BDL is a type of deep learning that utilizes Bayes' theorem to calculate the posterior distribution by incorporating prior distribution and likelihood of the data as input components [13]. This approach is useful in measuring different uncertainties related to the data and architecture used [13]. In BDL, various statistical distributions can be utilized to represent the prior and posterior distributions; however, the normal and Bernoulli distributions are the most prevalent when it comes to image data [14]. To generate output, the Bayesian approach obtains samples from the posterior distribution. Markov chain Monte Carlo (MCMC) is one of the most well-known exact sampling methods for posterior distribution, however, it is impractical to scale up for larger datasets, such as images, and can be computationally expensive [14]. To address this issue, other approximation methods such as variational inference (VI) [15] and Monte Carlo dropout (MC-Dropout) [16] are used. VI assumes that the data follows a normal distribution and samples from a normal distribution for the model's posterior [15]. On the other hand, MC-Dropout utilizes the Bernoulli distribution to generate a posterior distribution [16,17]. These methods are faster and have fewer parameters than MCMC, making them more feasible for larger-scale datasets [14].

This study aimed to assess the confidence level in predicting the results of deep learning models using BDL for the MLP-Mixer architecture. The focus of this research was on quantifying uncertainty in MLP-Mixers, rather than achieving optimal results. Furthermore, the study aimed to compare the uncertainty quantification of MLP-Mixers with that of several CNN architectures. Even though many works have been published on uncertainty quantification [13,14], to the best of our knowledge, this is the first study to apply BDL to estimate uncertainty in MLP-Mixers. Thus, this study provides essential insights into the potential and limitations of this architecture.

The rest of the paper is organized as follows: Section 2 presents the methods and data utilized in this study. Specifically, the section provides a detailed overview of the BDL approach and its application to uncertainty quantification in MLP-Mixers and CNN models. Section 3 reports the experimental results of the MLP-Mixers and CNN models. Section 4 presents a detailed discussion of uncertainty quantification in deep learning models with a particular focus on MLP-Mixers. Finally, Section 5 concludes the paper by summarizing the key findings and discussing their implications for future research in the field.

2. Methodology

In this study, BDL was applied to quantify the uncertainty of the MLP-Mixer model. To better understand the approach, it is crucial to understand the working mechanism of MLP-Mixer models. MLP-Mixer is a modern architecture used for vision tasks that uses multi-layer perceptrons (MLPs) as the fundamental building blocks. The idea behind MLP-Mixer is to replace convolutional layers with MLPs, which enables the network to have greater expressive power and flexibility. The MLP-Mixer architecture is composed of three primary components: patch processing, mixer layers, and classification head. In patch processing, the input image is initially divided into non-overlapping patches of size ($p \times p$), where p represents the patch size. Each patch is then flattened into a vector. The mixer layers are the key building blocks of MLP-Mixer. The output tokens of the last mixer layer are processed by a global average pooling layer to obtain a single feature vector. This vector is then passed through a fully connected layer, which produces the final classification output. The entire architecture of MLP-Mixers, as proposed by [9], is illustrated in Figure 1.

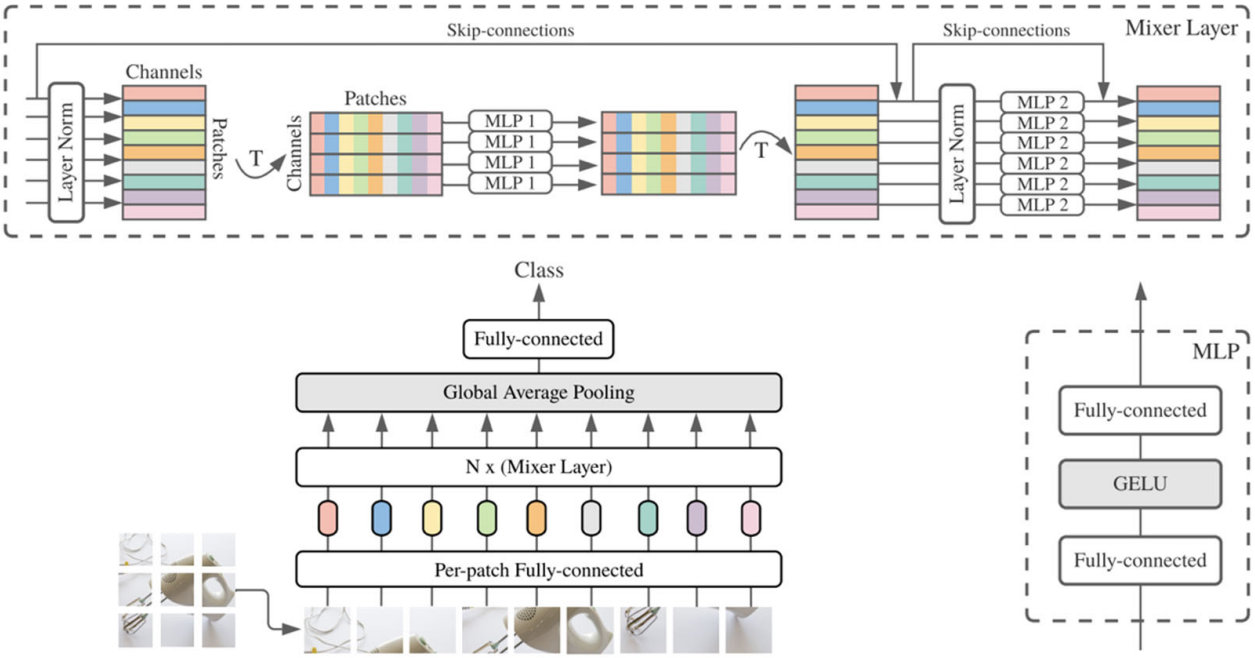


Figure 1. MLP-Mixer architecture used in ref. [9]. Reprinted from ref. [9].

In the MLP-Mixer model, each mixer layer consists of two sub-layers: token mixing and channel mixing.

- Token mixing: The token mixing layer applies a fully connected MLP to each patch vector, which generates a set of token vectors. These tokens represent the local information within each patch, and the MLP learns to transform the tokens to extract significant features, as mathematically given in Equation (1).

$$T' = (w_2\sigma(w_1(LN(X^*)) + b_1) + b_2) + X^* \quad (1)$$

  where T' is the output of the token mixing layer, $LN(X^*)$ is the layer normalization for each token vector, w_1 and b_1 are weight and biases for the first dense layer in token mixing, w_2 and b_2 are weight and biases for the second dense layer in token mixing, with a σ (GELU) nonlinearity activation function between the two dense layers. Finally, the X^* at the end represents the skip connection;
- Channel mixing: The channel mixing layer applies another MLP to the set of token vectors generated by the token mixing layer. This MLP is applied across the tokens of each patch, allowing global information to be captured across the entire image. The

output of the channel mixing layer is a new set of token vectors that are then fed to the next mixer layer. This is defined mathematically in Equation (2).

$$C' = \left(w_4\sigma\left(w_3\left(LN\left(T'\right)\right) + b_3\right) + b_4\right) + T' \quad (2)$$

where C' is the output of the channel mixing layer, $LN(T')$ is the layer normalization for each channel vector, w_3 and b_3 are weight and biases for the first dense layer in channel mixing, w_4 and b_4 are weight and biases for the second dense layer in channel mixing, with a σ (GELU) nonlinearity activation function between the two dense layers. Finally, the T' at the end represents the skip connection.

In this study, BDL was applied to the equations presented above to quantify the uncertainty in MLP-Mixer. Rather than using single point estimates for the weight and biases (w and b), the proposed method applied Bayesian distribution to both. However, in the utilized models, the GELU activation function from the original MLP-Mixer was replaced by ReLU in the utilized MLP-Mixers. The details of the materials and methods are presented in the following sections. Figure 2 depicts the process taken to quantify the uncertainty of models.

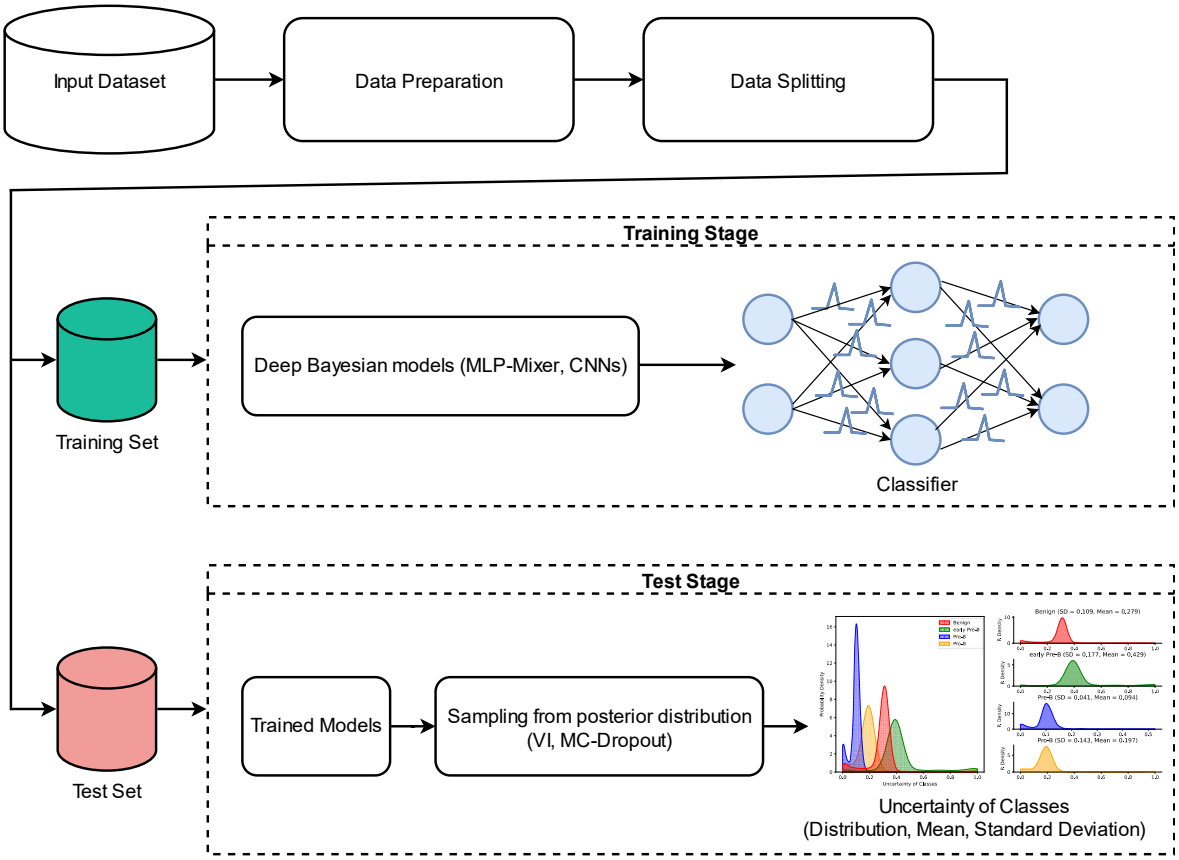


Figure 2. Flowchart of the utilized models and uncertainty quantification.

2.1. Datasets

This study utilized two relatively small datasets. The first dataset was characterized by low levels of noise, making it suitable for the classification task. The second dataset consisted of grayscale images that required different classification approaches compared to color images. Therefore, it was crucial to carefully select the appropriate dataset for this study. The two datasets are briefly explained in the following sections.

2.1.1. Acute Lymphoblastic Leukemia (ALL)

In this study, the ALL image dataset created by Mehrad et al. [18] was used to evaluate the proposed methods. Before discussing the technical details of the dataset, it is important to provide an overview of ALL.

ALL is a malignant neoplasm cancer that affects the blood and bone marrow. The bone marrow is the soft tissue inside the bones and is responsible for producing blood cells. ALL is characterized by the overproduction of immature white blood cells, known as lymphoblasts, that do not function properly and can outnumber normal blood cells. This can cause anemia, infections, and other serious health issues [19]. There are two main types of leukemia: acute and chronic. Acute leukemia is a more aggressive form that requires immediate treatment and is characterized by the rapid production of underdeveloped blood cells. Chronic leukemia, on the other hand, takes longer to develop and may not require immediate treatment [20]. ALL is a type of acute leukemia that affects lymphoid cells, which are a specific type of white blood cell that helps strengthen the immune system. ALL is more common in children than in adults and is identified by the rapid production of underdeveloped lymphocytes or lymphoblasts. This disease can be life-threatening if not treated promptly [21].

This study used a collection of images called the ALL dataset, which specifically focused on hematogenous, a type of underdeveloped white blood cell found in the bone marrow of children and adolescents. The dataset contains 3256 peripheral blood smear (PBS) images from 89 individuals suspected of having ALL, which were collected at Tehran's Taleqani Hospital in Iran [22]. The dataset was composed of 25 healthy individuals with benign conditions and 64 individuals who tested positive for a type of malignant lymphoblast. As stated by Mehrad et al. [18], the main purpose of this dataset was to classify images into four categories: benign, early Pre-B, Pre-B, and Pro-B, as illustrated in Figure 3. These categories are referring to different stages of cancer development in B-cells. B-cells are a type of white blood cell that play a role in the immune system. Benign refers normal, healthy lymphocytes that have not yet become cancerous B-cells, while pro-B refers to lymphocytes that are in the earliest stages of development that have not yet developed some of the key features of mature lymphocytes. Early Pre-B cells are lymphocytes that are in the early stages of development and have not yet fully matured. Pre-B cells are lymphocytes that are in a slightly more advanced stage of development and are closer to becoming fully mature. Early Pre-B and Pre-B are commonly affected by ALL. The images were enlarged to 100× using a Zeiss camera to ensure a suitable resolution for image processing. The reason for using this dataset in the current study is that it is relatively small, yet still allows a high level of accuracy to be achieved. Only a few researchers have used this dataset and CNNs to identify ALL using PBS, including Ghaderzadeh et al. [23], Atteia et al. [24], and Billah et al. [25]. Therefore, the proposed method in this study could provide additional insights and comparisons with previous research on this dataset.

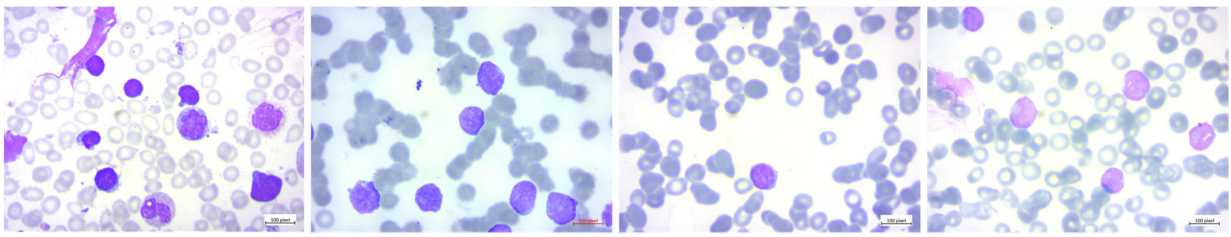


Figure 3. Samples from ALL dataset: (**A**) benign; (**B**) early Pre-B; (**C**) Pre-B; and (**D**) Pro-B.

2.1.2. Breast Cancer

Breast cancer is a type of cancer that originates in the breast tissue. It is among the most commonly occurring cancer type in women, although it may also occur in men, albeit less frequently. This disease occurs when abnormal cells within the breast tissue grow

uncontrollably and form a mass, referred to as a tumor. While some tumors are benign and not cancerous, others are malignant and can spread to other parts of the body if left untreated [26]. The diagnosis of breast cancer typically involves a combination of physical examination, imaging tests, such as mammography or ultrasound [27], and biopsy. Early detection and prompt treatment are key to increasing the chances of survival and recovery from breast cancer [28].

In 2018, Al-Dhabyani et al. [27], developed a dataset of medical images of breast cancer using ultrasound scans. The dataset, known as the Breast Ultrasound Dataset, is divided into three categories: normal, benign, and malignant images. With the help of machine learning algorithms, these images can be used for classifying, detecting, and segmenting breast cancer. The data consisted of 780 grayscale breast ultrasound images of 600 women aged between 25 and 75 years. The images were on average 500 × 500 pixels in size. This dataset was selected due to its relatively small size and grayscale format, which is beneficial for classification purposes in this study. Figure 4 illustrates a few samples from this dataset.

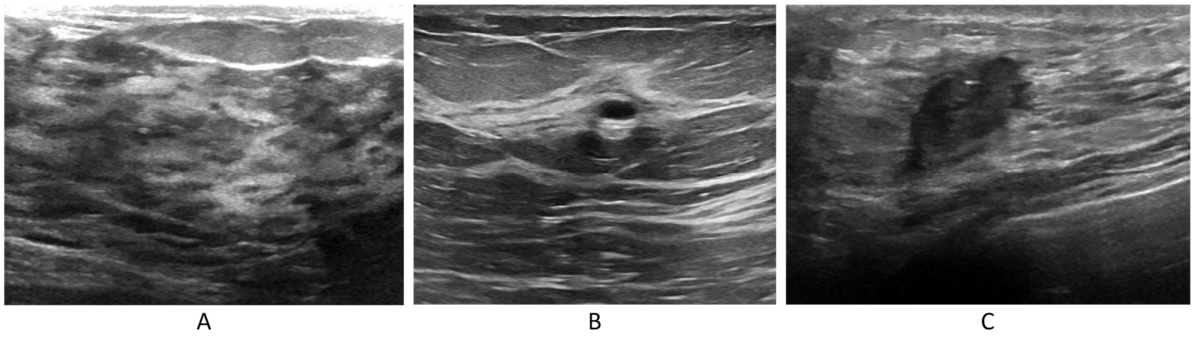


Figure 4. Samples from the breast cancer dataset: (**A**) normal; (**B**) benign; and (**C**) malignant.

2.2. Data Preparation

In this study, the data preparation process involved several steps aimed at optimizing the performance of the deep learning models. The first step involved resizing the images to 128 × 128 pixels to reduce computational requirements while maintaining an adequate level of resolution for accurate image analysis. The second step involved addressing the issue of class imbalance by assigning different weights to each class using a class weight, which is essential for ensuring the accurate classification of minority classes. However, this study did not employ any augmentation, resampling, or similar techniques, as the primary objective was to estimate uncertainty rather than achieve state-of-the-art results. While these methods might have improved the accuracy of our results, they would have introduced additional complexity and gone beyond the scope of our intended goals. Instead, the study focused on evaluating the inherent uncertainty present in the data, which provides valuable information for further analysis.

The evaluation of the model involved a three-fold data split where 15% of the data was allocated for testing, another 15% for validation, and the remaining 70% was used for training the models. Additionally, the data was normalized to ensure that features had similar scales, which is crucial for deep learning model stability and convergence. This approach to data preparation is essential for developing robust and generalizable deep learning models, which are important in many applications, including image analysis.

2.3. Uncertainty Quantification

BDL is a statistical approach that combines deep learning with Bayesian methods to estimate the uncertainty of model predictions. BDL estimates the probability distribution of model parameters based on observed data, allowing for the calculation of both point estimates and associated uncertainties. This method is particularly useful in classification problems, where the goal is to predict categorical labels given a set of inputs.

There are several methods to estimate uncertainty in BDL, but two popular methods are VI and MC-Dropout. Both methods are based on Bayes' theorem, which states that:

$$P(A|B) = \frac{P(A)\ P(B|A)}{P(B)} \tag{3}$$

where $P(A|B)$ is the posterior probability of A given B, $P(B|A)$ is the likelihood of B given A, $P(A)$ is the prior probability of A, and $P(B)$ is the marginal likelihood or evidence.

The predictive distribution can be calculated as:

$$P(y^*|x^*, D) = \int P(y^*|x^*, \omega)P(\omega|D)d\omega \approx \int P(y^*|x^*, \omega)q(\omega)d\omega \tag{4}$$

In VI, the assumption is that the true posterior distribution $P(w|D)$ (also denoted as $P(w|X, Y)$ in the context of machine learning) of the model parameter w can be approximated with a simpler distribution $q(w)$ that belongs to a tractable family of distributions. The goal of VI is to find the best approximation $q(w)$ by minimizing the Kullback–Leibler (KL) divergence between the true posterior distribution $P(w|D)$ and the approximate distribution $q(w)$ distribution:

$$KL(q(\omega)\|P(\omega|D) = \sum_{i=1}^{n} q_i(\omega).(\log q_i(\omega) - \log P_i(\omega|D)) \tag{5}$$

The KL divergence measures the distance between the two distributions and is non-negative, with a value of 0 only when the two distributions are identical. Therefore, minimizing the KL divergence is equivalent to finding the best approximation $q(w)$ that is closest to the true posterior distribution $P(w|D)$.

However, the MC-Dropout method involves using dropout during both training and testing to estimate the model's uncertainty. During training, dropout is applied to the inputs and/or hidden units of the neural network with a certain probability. This creates an ensemble of different neural networks that share weights, with some neurons randomly dropped out. The loss function is computed for each dropout run, and the weights are updated based on the average of the losses. During testing, the model is run multiple times with dropout applied, and the predictions are averaged over the runs. The dropout rate is considered a hyperparameter that controls the model's uncertainty. The final predictions are given by:

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

where $p(y^*|x^*, \omega)$ is the predicted probability of class i for the t th dropout run, and T is the number of dropouts runs. This method can be viewed as an approximation of Bayesian model averaging, where the dropout masks are treated as different models, and the predicted probabilities are averaged to estimate the expected probability of each class given the input x^*. After computing the predicted probabilities for each class using MC-Dropout, the final predicted class is obtained by applying the argmax function to the predicted probabilities. The argmax function returns the index of the class with the highest predicted probability.

Both VI and MC-Dropout are effective approaches that can enhance the reliability and resilience of deep learning models. VI provides a probabilistic framework for estimating complex posterior distributions, allowing for more accurate uncertainty estimations for individual neurons. On the other hand, MC-Dropout provides a computationally efficient way to estimate model uncertainty through the use of dropout masks during both training and testing. Nevertheless, both methods have their own strengths and limitations, and the choice of which one to use will depend on the specific problem and the available computational resources. For instance, VI can be computationally demanding since it in-

volves calculating the gradients of the lower bound objective with respect to the variational parameters. In contrast, MC-Dropout may not produce precise results if the number of dropout runs is small, and it may not be suitable for problems where the uncertainty is highly dependent on the input. Therefore, selecting either VI or MC-Dropout necessitates careful consideration of the specific problem requirements, the computational resources available, and the balance between accuracy and efficiency.

3. Experiments and Results

To evaluate the effectiveness of the proposed method, an empirical study was conducted using five deep learning models based on MLP-Mixers and CNN architectures. The models were trained and tested on two different image datasets to assess their performance in image classification tasks. Two versions of each model were tested: one with MC-Dropout and the other with VI. The models were trained for varying numbers of epochs with early stopping criteria. However, most of the models lasted between 30 and 80 epochs, with the main difference being that MC-Dropout models tended to stop earlier than VI models in general.

The first model utilized in this study was the MLP-Mixer model, which employed a unique approach to image classification. In this model, the input images were divided into 32 patches, and a custom MLP-Mixer was designed for this study. The custom MLP-Mixer had approximately 30 million trainable parameters for both the MC-Dropout and VI versions of the model. Unlike many other models, the MLP-Mixer was not pre-trained on any dataset, as it was developed specifically for this study. Furthermore, it was built entirely from dense layers, without incorporating any CNN layers. This was a deliberate design choice, as the researchers sought to investigate the potential of dense layer networks for image classification tasks.

The second model employed in this study was a custom CNN model composed of four CNN layers to extract features from the input images. The CNN layers consist of multiple filters that perform feature extraction operations such as detecting edges, textures, and shapes. The extracted features were then fed into three dense layers, including an output layer with four/three classes for the two given datasets. In this output layer, a softmax activation function was used to produce a probability distribution over the four classes. Notably, the custom CNN contained around 38 million trainable parameters for both MC-Dropout and VI models.

The third model used in the experiments was the ResNet152, which is a widely used and popular model for image classification. This model consists of 152 convolutional layers and was pre-trained on the large-scale ImageNet dataset. The ResNet152 was modified for the classification task in this study by adding three dense layers, with the final layer being an output layer that employed a softmax activation function to generate a probability distribution over the classes. The utilized ResNet had around 60 million trainable parameters for both MC-Dropout and VI models.

The fourth model utilized in the experiments is based on the Inception V3 model, which is a pre-trained model that has previously been trained on the ImageNet dataset. This model is well-known for its remarkable ability to detect and classify objects in images, making it a suitable choice for this study. The Inception V3 model has around 42 layers, and three dense layers were added to the top of the pre-trained model. The output layer consisted of four/three classes for the utilized datasets, and a softmax activation function was employed to produce a probability distribution over these classes. Both MC-Dropout and VI models of the Inception V3 model had approximately 23 million parameters.

The last model used in our experiments is based on the DenseNet121 architecture, which is an efficient CNN architecture designed to mitigate the problem of vanishing gradients and improve the flow of gradients. This architecture consists of dense blocks, where each dense block contains multiple layers connected by direct connections, allowing the model to use the features learned by previous layers and learn more complex representations. The DenseNet121 is pre-trained on the ImageNet dataset and has a total of

120 convolution layers. Three dense layers, including an output layer with four/three classes, were added on top of the model, and a softmax activation function was used to produce a probability distribution over the four classes. The number of trainable parameters was around 8 million for both MC-Dropout and VI models.

The last 10 layers of the ResNet, Inception, and DenseNet models were fine-tuned using the two datasets. Fine-tuning the pre-trained models on the utilized datasets was implemented as a means of comparing their performance with that of the custom model, as well as the potential benefit of fine-tuning the latter on the utilized datasets used in this study.

For all the deep learning models, the Adam optimizer was used during the training process. The Adam optimizer is a well-known and efficient optimization algorithm for deep learning problems and is known for its rapid convergence. We used the ReLU activation function, which is known for its ability to introduce nonlinearity into a model and allow it to learn more complex representations of the data. To estimate the test data uncertainty, we took a total of 100 samples for each model.

The objective of this study was to investigate and differentiate the uncertainty present in various deep learning models by using MC-Dropout and VI methods. To achieve this objective, a comprehensive comparison of the two methods was conducted, and the results were meticulously analyzed. The results obtained from the models studied are presented in Tables 1 and 2 for the two datasets to facilitate a clear and concise comparison. This study highlighted the significance of incorporating uncertainty into the development of machine learning models.

Table 1. Resultfootnotes of uncertainty quantification for different deep learning models used for the ALL dataset.

Model	Mean Class Confidence & Standard Deviation	Classes				Accuracy of Model	
		Benign	Early Pre-B	Pre-B	Pro-B	Base Model	Sampling
Custom CNN + MC-Dropout	Mean SD	0.9031 0.1546	0.9356 0.1210	0.9809 0.0778	0.9978 0.0178	0.8973	0.8973
Custom CNN + VI	Mean SD	0.8442 0.1509	0.7915 0.1369	0.9246 0.0741	0.9178 0.0432	0.8603	0.9178
Inception + MC-Dropout	Mean SD	0.9388 0.1237	0.9167 0.1357	0.9837 0.0644	0.9661 0.1038	0.9425	0.9404
Inception + VI	Mean SD	0.8511 0.1472	0.8856 0.1081	0.9202 0.0969	0.9081 0.0983	0.8829	0.9404
DenseNet + MC-Dropout	Mean SD	0.9825 0.0600	0.9764 0.0795	0.9949 0.0307	0.9871 0.0508	0.9774	0.9712
DenseNet + VI	Mean SD	0.9263 0.0994	0.9141 0.1185	0.9078 0.1122	0.9551 0.0525	0.9445	0.9568
ResNet + MC-Dropout	Mean SD	0.7680 0.1973	0.8986 0.1393	0.9415 0.1220	0.9552 0.1148	0.6201	0.9507
ResNet + VI	Mean SD	0.4873 0.1238	0.5966 0.1388	0.6924 0.1749	0.6944 0.1545	0.4065	0.7351
MLP-Mixer + MC-Dropout	Mean SD	0.8448 0.1683	0.9179 0.1425	0.9664 0.0936	0.9542 0.1037	0.8706	0.9507
MLP-Mixer + VI	Mean SD	0.7626 0.2252	0.8844 0.1346	0.9671 0.0867	0.8852 0.0952	0.8603	0.9466

Table 2. Results of uncertainty quantification for different deep learning models used for the breast cancer dataset.

Model	Mean Class Confidence & Standard Deviation	Classes			Accuracy of Model	
		Normal	Benign	Malignant	Base Model	Sampling
Custom CNN + MC-Dropout	Mean SD	0.9192 0.1481	0.9178 0.1217	0.9569 0.0570	0.7264	0.7179
Custom CNN + VI	Mean SD	0.8927 0.1511	0.9011 0.1264	0.8921 0.1363	0.7350	0.7778
Inception + MC-Dropout	Mean SD	0.9019 0.1511	0.9034 0.1455	0.9383 0.1418	0.7777	0.7692
Inception + VI	Mean SD	0.8882 0.1411	0.8223 0.1739	0.8279 0.1642	0.7521	0.7435
DenseNet + MC-Dropout	Mean SD	0.9337 0.1151	0.9245 0.0993	0.9330 0.1078	0.8290	0.8547
DenseNet + VI	Mean SD	0.9061 0.1447	0.8418 0.1685	0.8801 0.1327	0.7692	0.7607
ResNet + MC-Dropout	Mean SD	0.8951 0.1420	0.7149 0.1812	0.8194 0.1466	0.5299	0.7009
ResNet + VI	Mean SD	0.8313 0.1299	0.7645 0.1451	0.7947 0.1578	0.6324	0.7692
MLP-Mixer + MC-Dropout	Mean SD	0.7566 0.2068	0.7925 0.2069	0.7880 0.2333	0.6239	0.6923
MLP-Mixer + VI	Mean SD	0.6598 0.2188	0.6795 0.1467	0.7417 0.1803	0.5384	0.6325

The results displayed in Table 1 indicate that MC-Dropout achieves higher accuracy than VI. The mean and standard deviation of each class shows a higher level of confidence with MC-Dropout than with VI. The decrease in accuracy with VI also indicates a lower level of confidence in its predictions. This suggests that MC-Dropout is better at identifying the true class probabilities, while VI tends to overgeneralize and make predictions with less certainty. These findings are crucial in determining the appropriate model for different tasks and applications. Interestingly, the MLP-Mixer performed better than other models, except for DenseNet, while it was the second-worst model before sampling. This indicates that BDL can significantly improve the performance of MLP-Mixer. Additionally, the mean and standard deviation of each class provides insights into which classes the model performs better or worse on.

Table 2 presents the results of the second dataset, which was a relatively small dataset. Despite the dataset's limited size, all models were able to efficiently capture the uncertainty associated with both the model and the data. The results indicate that the MC-Dropout approach tends to be more confident in its predictions compared to the VI version of the same model. This is evident in the higher mean and lower standard deviation of the predicted classes overall. Although the VI versions showed better accuracy in some cases, MC-Dropout demonstrated a higher level of confidence. As expected, the MLP-Mixers performed the worst in terms of accuracy. However, their performance improved significantly when BDL sampling was utilized, especially with the MC-Dropout approach. This finding is of particular interest as it suggests that MLP-Mixers can benefit greatly from incorporating uncertainty quantification. Therefore, this study highlights the importance of considering both accuracy and uncertainty when evaluating machine learning models and provides valuable insights into the potential of MLP-Mixers in real-world applications.

Our experimental findings reveal a noticeable improvement in the performance of MLP-Mixers by utilizing BDL to estimate their uncertainty. Although the MLP-Mixer did not achieve state-of-the-art results, our findings suggest that with some modifications, it has the potential to further enhance its performance. Additionally, we used a basic architecture for the mixer, demonstrating its versatility. With the availability of other mixer variants, we are optimistic that by combining them with BDL, mixer models can effectively compete with state-of-the-art approaches, even on small to medium-sized datasets. These results

offer promising directions for future research and highlight the potential of MLP-Mixers with BDL for practical applications in various domains.

Figures 5 and 6 display a few examples of uncertainty quantification for the ALL dataset using the MC-Dropout approach. To accurately quantify the uncertainty of the predicted classes, 1000 samples were taken for each example. It should be noted that these image examples were randomly selected, which means that image (A) in Figure 5 is different from (A) in Figure 6, and similarly for (B) and (C). The (A) in both Figures 5 and 6 represents an accurately classified example with very low uncertainty (high confidence). We observed that when making predictions with low uncertainty, the mean for the true class was very close to 1.0, while the mean for other classes was close to 0.0. In addition, the standard deviation was very low for both true and false classes. Image (B) in both Figures 5 and 6 depicts a truly predicted class with moderately high uncertainty. Interestingly, the CNN model tended to have a curve peak near both ends of the scale, while the MLP-Mixer behaved very differently, with the curve peak at any point on the scale. Additionally, in general, the MLP-Mixer demonstrated higher uncertainty compared to CNN models, which is reflected in the results from Tables 1 and 2. Image (C) in both Figures 5 and 6 represents a misclassified category. In Figure 5C, even though the class "early Pre-B" is not the actual label for the associated image, it has a peak close to 1.0. Although this is an extreme case, it shows that CNN models generally have a peak in the curve near the beginning and the end of the scale. In Figure 6C, a very different pattern emerges, where the curves are selected at random points instead of at the beginning or the end of the scale. Notably, Figure 6C for MLP-Mixer has a lower mean for the incorrectly classified label compared to Figure 5C of the CNN model.

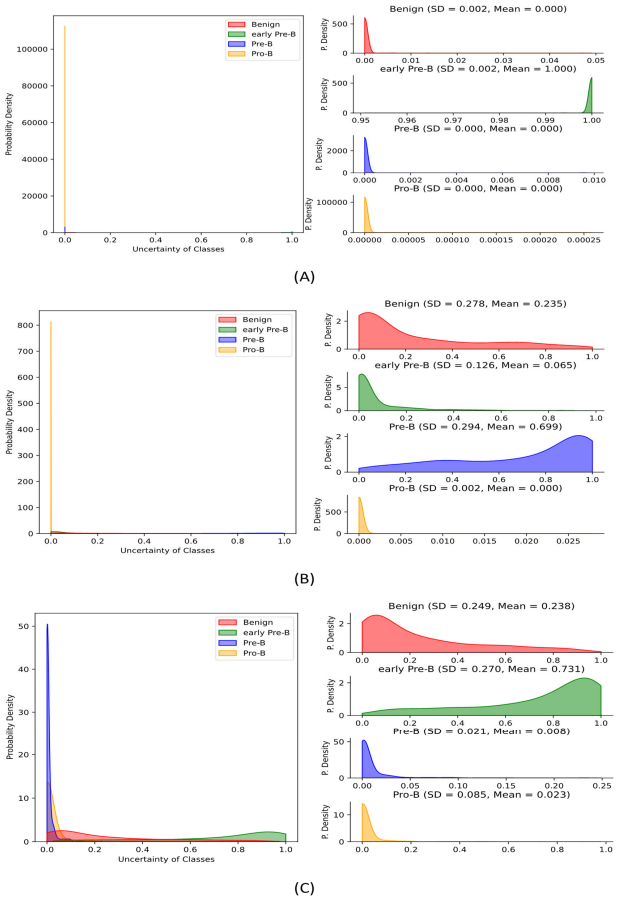


Figure 5. Examples of uncertainty quantification for ALL dataset using custom CNN architecture: (**A**) correctly classified with low uncertainty; (**B**) correctly classified with high uncertainty; and (**C**) misclassified with high uncertainty.

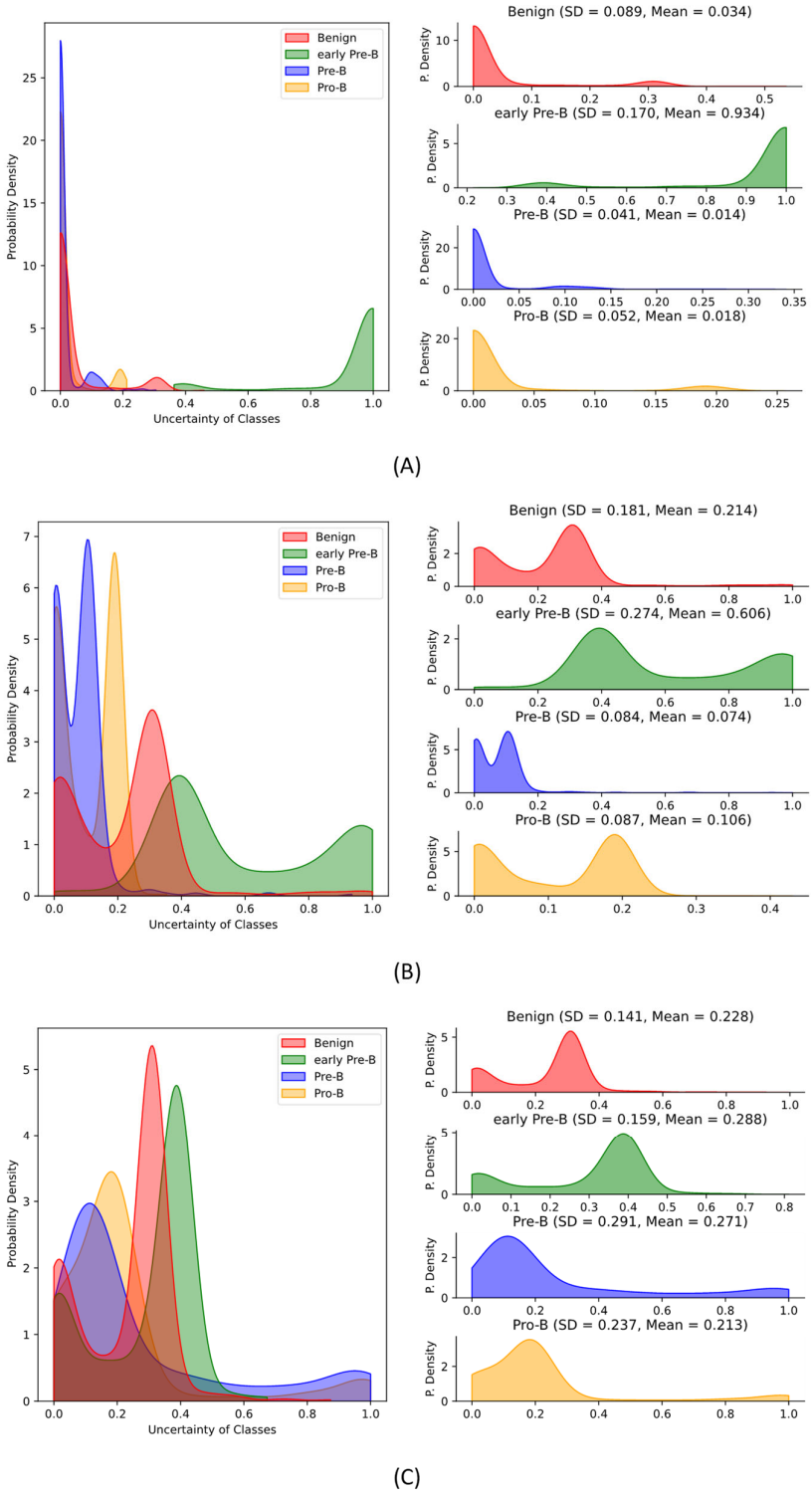


Figure 6. Examples of uncertainty quantification for ALL dataset using custom MLP-Mixer architecture: (**A**) correctly classified with low uncertainty; (**B**) correctly classified with high uncertainty; and (**C**) misclassified with high uncertainty.

In addition to the above figures, we conducted further experiments on both datasets and arrived at several more conclusions. First, we observed that MLP-Mixers tend to have a lower mean and a higher standard deviation most of the time, indicating lower confidence in their predictions in general. However, this can actually be beneficial because it means that when the model has low uncertainty, we can be more confident in the accuracy of the

prediction. Second, we found that MLP-Mixers tend to have more random curves that can have a peak at any point on the scale, unlike CNN architectures that typically have peaks near the beginning and end of the scale. Additionally, both CNN-based architectures and MLP-Mixers can exhibit high uncertainty for misclassified labels, but MLP-Mixer shows higher uncertainty in most cases. These findings further highlight the importance of considering both accuracy and uncertainty when evaluating machine learning models.

4. Discussion

The results presented in this study suggest that incorporating uncertainty quantification techniques, specifically BDL, can significantly enhance the performance of MLP-Mixers. This approach not only improves the model's accuracy but also enables quantification of the uncertainty associated with both the models and the data. However, it is crucial to note that while the results of this study have been compared with those of CNN models, there are other factors to consider when using uncertainty quantification that has not been explored in this research. To provide a more comprehensive understanding of the use of uncertainty quantification, it is essential to consider the following additional points.

- In the context of BDL, it is frequently observed that a model's performance declines when the mean of the predicted true classes falls below a certain threshold, which in this study's datasets was around 50%. This situation is characterized by a decrease in the model's standard deviation, indicating a failure to capture the associated uncertainty. It is worth noting that this observation does not necessarily imply a failure of the BDL, but rather indicates that the model has failed to effectively generalize the results and predict each class accurately. In such cases, it is recommended to explore alternative methods or models that may be more suitable to the data and task at hand. This approach may involve adjusting the model's hyperparameters or selecting a different model architecture altogether. Additionally, it may be necessary to reconsider the data preprocessing steps or feature engineering techniques used in the model development process. Therefore, it is important to note that the identification of poor-performing models and the subsequent exploration of alternative approaches are crucial steps in the iterative model development process. It is through this iterative process that one can gain a better understanding of the data, model, and associated uncertainties, ultimately leading to improved model performance and increased confidence in the resulting predictions;
- In BDL, when a model is presented with unfamiliar data, the associated posterior distributions become more uncertain, resulting in higher uncertainty estimates. This increased uncertainty indicates that the network is less confident in its predictions, and it can serve as a useful tool for identifying abnormal or unusable data that significantly deviate from the training dataset. By utilizing the model uncertainty estimates to identify such data, BDL models, particularly MLP-Mixer models, can deliver improved performance and more reliable results across various applications. This approach to leveraging uncertainty estimates to detect abnormal data aligns with the wider concept of model-based anomaly detection, which has proven successful in various applications. Additionally, by quantifying the associated uncertainty, BDL models can provide a valuable tool for identifying data points that are on the edge of the model's training distribution. This can serve as a foundation for further exploring the data-generating process and improving the model's overall generalization ability;
- It is important to understand that a perfect score on a dataset does not necessarily imply perfect confidence or zero uncertainty in a model's predictions. Our research findings, as presented in Tables 1 and 2, demonstrate this fact. However, some publications have misinterpreted the relationship between a perfect score and uncertainty. It is important to note that the likelihood of a model having zero uncertainty or standard deviation is extremely low and that this situation can only occur under extreme circumstances. This highlights the fact that even when a model performs exceptionally well on a dataset, some degree of uncertainty remains in its predictions due to various factors

such as model bias, incomplete information, and data variability. It is, therefore, essential to consider the uncertainty associated with a model's predictions, as it can provide valuable insights into the reliability of its results and potential areas for improvement. Failure to account for such uncertainty can lead to overconfidence in a model's predictions, which can be detrimental in critical applications such as healthcare. Researchers and practitioners must be aware of the presence of uncertainty and focus on developing methods to quantify and account for it. By doing so, we can improve the overall reliability and robustness of deep learning models and ensure their effective application across a wide range of fields;

- In the field of machine learning, it is commonly held that using fully Bayesian methods in deep learning might not always lead to better results compared to approximate methods. This is due to the fact that fully Bayesian methods may not be effective with all types of data, particularly in the case of image data, where images can vary significantly from one another. In addition, using fully Bayesian methods can be computationally demanding, especially with models that have a large number of parameters to learn. Therefore, when deciding between fully Bayesian and approximate methods, it is important to consider factors such as the size of the data, the computing resources available, and the desired level of model complexity. Considering these factors can aid in selecting the most appropriate method for each specific task. Consequently, full BDL has not received widespread acceptance among machine learning researchers, particularly in the field of computer vision;
- The MLP-Mixer is known for its relatively limited interaction with neighboring neurons compared to CNNs. This limited interaction can result in decreased performance when working with smaller datasets, as the model may struggle to capture the complex patterns and relationships present in the data. However, for very large datasets, MLP-Mixer can perform equally well as CNN models while being more computationally efficient. Therefore, it is important to carefully consider the strengths and limitations of MLP-Mixers when choosing an architecture for a given task, especially when working with smaller datasets or in environments with limited computational resources. Researchers have developed different versions of MLP-Mixer to increase interaction between neighboring regions while decreasing interaction between further away regions [10]. Our findings suggest that incorporating uncertainty quantification using BDL can significantly enhance the performance of MLP-Mixers, particularly for small and medium-sized datasets;
- Unlike CNN models that use filters for uncertainty quantification, MLP-Mixers utilize regions of images or feature maps, offering a more comprehensive analysis of the data. This approach provides a more accurate representation of uncertainty, which can lead to better performance in certain applications;
- Although BDL often results in high estimated uncertainty for incorrectly assigned classes, it may also produce high uncertainty for correctly assigned classes as well. This phenomenon is particularly common among classes that are closely related. For example, in the context of cancer classification (Table 2), the normal and benign classes may both have high uncertainty, while the malignant class has low uncertainty. Alternatively, the benign and malignant classes may both exhibit high uncertainty, while the normal class has low uncertainty. However, it is less common for both the normal and malignant classes to show high uncertainty, while the benign class has low uncertainty, as these two classes possess noticeably distinct features that distinguish them from each other. Therefore, understanding the underlying factors that contribute to the uncertainty estimates of a model is crucial for interpreting and evaluating its performance. Additionally, taking into account the inherent uncertainty associated with a model's predictions can provide valuable insights into the reliability and robustness of its results, especially in applications where misclassification can have severe consequences.

5. Conclusions

This paper presented a study on the use of uncertainty quantification in MLP-Mixers for classifying small datasets with BDL. The research compared two versions of MLP-Mixers that used MC-Dropout and VI with four different CNN models. Similarly, two versions of MC-Dropout and VI were implemented for each CNN model. The objective of this study was to investigate the effectiveness of uncertainty quantification in MLP-Mixers, rather than achieving state-of-the-art results. Two small medical datasets, which differed in various criteria, were used to evaluate, and compare, the performance of the MLP-Mixers with the other CNN models. The results of the study revealed that the MLP-Mixers did not surpass all the other models together; however, uncertainty quantification could greatly improve the performance of MLP-Mixers, particularly when using MC-Dropout. This was not consistently observed with the CNN models. The study highlights the importance of considering uncertainty quantification in MLP-Mixers, especially when working with small datasets. Future research should investigate the potential of preprocessing techniques, such as augmentation and resampling, to enhance the performance of uncertainty quantification in MLP-Mixers models.

Author Contributions: Conceptualization, A.A.A., M.M.H. and Y.T.M.; methodology, A.A.A.; software, A.A.A.; validation, A.A.A., M.M.H. and Y.T.M.; writing—original draft preparation, A.A.A.; writing—review and editing, M.M.H. and Y.T.M.; supervision, M.M.H. and Y.T.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This work has utilized two publicly available datasets. The first dataset is Acute Lymphoblastic Leukemia which can be found here: https://www.kaggle.com/datasets/mehradaria/leukemia (accessed on 9 March 2023). The second dataset is Breast Ultrasound Images Dataset which can be found here: https://www.kaggle.com/datasets/aryashah2k/breast-ultrasound-images-dataset (accessed on 9 March 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bhatt, D.; Patel, C.; Talsania, H.; Patel, J.; Vaghela, R.; Pandya, S.; Modi, K.; Ghayvat, H. CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope. *Electronics* **2021**, *10*, 2470. [CrossRef]
2. Zhang, A.; Lipton, Z.C.; Li, M.; Smola, A.J. Dive into Deep Learning. *arXiv* **2021**, arXiv:2106.11342.
3. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions. *J. Big Data* **2021**, *8*, 53. [CrossRef] [PubMed]
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
5. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
6. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
7. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image Is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
8. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–17 October 2021; pp. 9992–10002.
9. Tolstikhin, I.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. MLP-Mixer: An All-MLP Architecture for Vision. *Adv. Neural Inf. Process Syst.* **2021**, *34*, 24261–24272.
10. Liu, R.; Li, Y.; Tao, L.; Liang, D.; Zheng, H.-T. Are We Ready for a New Paradigm Shift? A Survey on Visual Deep MLP. *Patterns* **2022**, *3*, 100520. [CrossRef]

11. Song, B.; Sunny, S.; Li, S.; Gurushanth, K.; Mendonca, P.; Mukhia, N.; Patrick, S.; Gurudath, S.; Raghavan, S.; Tsusennaro, I.; et al. Bayesian Deep Learning for Reliable Oral Cancer Image Classification. *Biomed. Opt. Express.* **2021**, *12*, 6422. [CrossRef]
12. Kendall, A.; Gal, Y. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: Red Hook, NY, USA, 2017.
13. Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U.R.; et al. A Review of Uncertainty Quantification in Deep Learning: Techniques, Applications and Challenges. *Information Fusion* **2021**, *76*, 243–297. [CrossRef]
14. Abdullah, A.A.; Hassan, M.M.; Mustafa, Y.T. A Review on Bayesian Deep Learning in Healthcare: Applications and Challenges. *IEEE Access* **2022**, *10*, 36538–36562. [CrossRef]
15. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational Inference: A Review for Statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877. [CrossRef]
16. Gal, Y.; Ghahramani, Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of the 33rd International Conference on Machine Learning, ICML, New York, NY, USA, 20–22 June 2016; pp. 1050–1059.
17. Wu, A.; Nowozin, S.; Meeds, E.; Turner, R.E.; Hernández-Lobato, J.M.; Gaunt, A.L. Deterministic Variational Inference for Robust Bayesian Neural Networks. In Proceedings of the 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019.
18. Aria, M.; Ghaderzadeh, M.; Bashash, D.; Abolghasemi, H.; Asadi, F.; Hosseini, A. Acute Lymphoblastic Leukemia (ALL) Image Dataset. *Kaggle* **2021**. [CrossRef]
19. Mahmood, N.; Shahid, S.; Bakhshi, T.; Riaz, S.; Ghufran, H.; Yaqoob, M. Identification of Significant Risks in Pediatric Acute Lymphoblastic Leukemia (ALL) through Machine Learning (ML) Approach. *Med. Biol. Eng. Comput.* **2020**, *58*, 2631–2640. [CrossRef]
20. Hafeez, M.U.; Ali, M.H.; Najib, N.; Ayub, M.H.; Shafi, K.; Munir, M.; Butt, N.H. Ophthalmic Manifestations of Acute Leukemia. *Cureus* **2019**, *11*, e3837. [CrossRef]
21. Rafei, H.; Kantarjian, H.M.; Jabbour, E.J. Recent Advances in the Treatment of Acute Lymphoblastic Leukemia. *Leuk. Lymphoma* **2019**, *60*, 2606–2621. [CrossRef]
22. Ghaderzadeh, M.; Asadi, F.; Hosseini, A.; Bashash, D.; Abolghasemi, H.; Roshanpour, A. Machine Learning in Detection and Classification of Leukemia Using Smear Blood Images: A Systematic Review. *Sci. Program.* **2021**, *2021*, 1–14. [CrossRef]
23. Ghaderzadeh, M.; Aria, M.; Hosseini, A.; Asadi, F.; Bashash, D.; Abolghasemi, H. A Fast and Efficient CNN Model for B-ALL Diagnosis and Its Subtypes Classification Using Peripheral Blood Smear Images. *Int. J. Intelligent Syst.* **2022**, *37*, 5113–5133. [CrossRef]
24. Atteia, G.; Alhussan, A.; Samee, N. BO-ALLCNN: Bayesian-Based Optimized CNN for Acute Lymphoblastic Leukemia Detection in Microscopic Blood Smear Images. *Sensors* **2022**, *22*, 5520. [CrossRef]
25. Billah, M.E.; Javed, F. Bayesian Convolutional Neural Network-Based Models for Diagnosis of Blood Cancer. *Appl. Artif. Intell.* **2022**, *36*, 2011688. [CrossRef]
26. Siegel, R.L.; Miller, K.D.; Fuchs, H.E.; Jemal, A. Cancer Statistics, 2022. *CA Cancer J. Clin.* **2022**, *72*, 7–33. [CrossRef]
27. Al-Dhabyani, W.; Gomaa, M.; Khaled, H.; Fahmy, A. Dataset of Breast Ultrasound Images. *Data Brief* **2020**, *28*, 104863. [CrossRef]
28. Nassif, A.B.; Talib, M.A.; Nasir, Q.; Afadar, Y.; Elgendy, O. Breast Cancer Detection Using Artificial Intelligence Techniques: A Systematic Literature Review. *Artif. Intell. Med.* **2022**, *127*, 102276. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.