Deep Bayesian Active Learning with High-Dimensional Data – A Critical Review

Nicolas Malz (Student ID: 12516149)
Contact: n.malz@campus.lmu.de
Paper for the Seminar 'Machine Learning with Limited Labels'
Summer Term 2024, *LMU Munich*

***Abstract*—This paper presents the deep active learning framework using Bayesian CNNs for image data as presented by Gal et al. By integrating Monte Carlo dropout for approximate Bayesian inference, the approach quantifies the model's epistemic uncertainty which is essential for active learning. Acquisition functions like BALD, Max Entropy, and Variation Ratios are evaluated in the Bayesian active learning setting, exploring their superiority over deterministic baseline methods. However, the Bayesian approach falls short of modern ensemble-based approaches' performance on both benchmark and real-world datasets.**

***Index Terms*—bayesian deep learning, image classification, deep learning, convolutional neural networks, active learning, BALD, ensemble learning**

I. Introduction

Deep learning and active learning are important subfields of machine learning. Deep learning utilizes artificial deep neural networks to automatically extract features from data, enabling complex learning capabilities through many simpler structures [1], [2]. The advent of large-scale annotated datasets and advancements in computational power have significantly accelerated deep learning research, giving it a performance edge over traditional machine learning algorithms in applications like high-dimensional classification and generative tasks [3]. However, deep learning's flexibility necessitates substantial amounts of labeled data for effective training. Data labeling in turn is inherently costly, labor-intensive, and time-consuming.

Active learning, also known as query learning, aims for efficient model updates by evaluating the value of different samples within a dataset. It starts with a small labeled training set and uses an acquisition function to determine which data points from the unlabeled pool should be labeled by an 'oracle' (typically a human expert) and added to the training set. The model is retrained on the updated training set, and this process is repeated until the desired model performance is achieved. Active learning minimizes the number of samples to be labeled, thus reducing costs while maximizing performance gains [4]. This is particularly beneficial in domains where labeling data is expensive and time-consuming, such as medical imaging or remote sensing.

Combining the strengths of deep learning and active learning, deep active learning has emerged, offering both extensive feature extraction capabilities and efficient sampling. Despite their potential, deep active learning algorithms have struggled with high-dimensional data, limiting their use to lower-dimensional problems and algorithms for a long time [4], [5]. Consequently, effective integration of deep neural networks into the active learning paradigm remains an active research field, especially for higher dimensional problems such as learning from images or natural language processing (NLP).

The challenges of using deep learning in active learning settings arise from two main issues: 1) Deep learning algorithms require large amounts of training data, whereas active learning demands we begin with a small training set; 2) Active learning acquisition functions need to quantify a model's uncertainty over unseen data (epistemic uncertainty). However, classic deterministic neural networks typically provide point estimates of uncertainty, which predominantly contain aleatoric uncertainty, i.e. the inherent noise in the prediction, rather than the epistemic uncertainty needed for effective active learning. Classic neural networks' uncertainty measures (e.g. softmax layer) are therefore less suited for active learning settings.

The lack of epistemic uncertainty quantification in traditional neural networks is inherent and arises from their training process. Neural networks are typically trained using backpropagation to calculate the gradients of a loss function (e.g., mean squared error for regression or cross-entropy for classification), followed by gradient descent to update the network's parameters [1]. This process aims to find a single set of parameters that minimizes the loss function, resulting in point estimates of epistemic uncertainty rather than the more informative probability distributions over possible parameters. Consequently, a trained neural network's parameters are fixed values that represent the best estimate given the training data [3]. Classic neural networks' deterministic nature strictly implies that the same input will always yield the same output, with no variation to indicate epistemic uncertainty in the prediction. Estimating epistemic uncertainty in neural networks would require capturing the variability or confidence in their predictions.

This paper presents a deep active learning approach by Gal et al. [6], which integrates Convolutional Neural Networks (CNNs), Bayesian probability theory, and active learning into a robust framework for high-dimensional data. By leveraging Bayesian CNNs and Monte Carlo (MC) dropout, this method effectively captures epistemic model uncertainty, selecting the most informative data points and improving model performance with fewer labeled samples. Specifically, the aim is to address the following research questions in the Bayesian active learning setting for image classification: 1) Which acquisition functions are most effective in selecting informative samples? 2) How does the performance of Bayesian CNNs compare to traditional deterministic CNNs and existing active or semi-supervised learning techniques in providing reliable uncertainty estimates? 3) What are the practical implications of using deep Bayesian active learning and its rival approaches in real-world applications?

The rest of this paper is structured as follows: Section II reviews related work in active learning and Bayesian deep learning, highlighting challenges and previous attempts to integrate these approaches. Section III details the methodology, including the architecture of Bayesian CNNs, the implementation of MC dropout, and the design of acquisition functions for deep active learning. Section IV presents the experimental setup and results, comparing the proposed method against traditional and state-of-the-art ensemble techniques on benchmark datasets. The practical implications, including applications to real-world problems like medical image analysis, are also discussed. Finally, Section VI concludes the paper and outlines potential directions for future research, focusing on improving computational efficiency and extending the framework to other types of high-dimensional data.

II. Related Work

At the time of Gal et al.'s paper's publication in 2017, research on tackling high-dimensional machine learning problems through active learning approaches was scarce. This section will first review the work available to the authors at the time before looking at more recently emerged paradigms and methods.

A. *Review of Related Work at the Time of Publication*

First attempts at expanding active learning to high-dimensional data were continuations of lower dimensional approaches: For instance, Joshi et al. [7] used probabilistic outputs derived from a Support Vector Machine (SVM) [8] with linear, polynomial, and Radial Basis Function (RBF) kernels on images to handle non-linearities. They employed entropy and Best-versus-Second Best (BvSB) as the primary measures of uncertainty to select the most informative examples for labeling. While their approach demonstrated effectiveness in reducing labeled data requirements, it faced challenges in scaling to the complexities of high-dimensional data typically found in modern applications.

Later attempts at scaling active learning utilized Gaussian Processes (GPs), a non-parametric approach to modeling data distributions, again with RBF kernels to obtain model uncertainty on top of predictions [9]. An important limitation of this approach, however, was only using (low-dimensional) scale-invariant feature transform (SIFT) features of the images as input. Finally, Zhu et al. [10] used a Gaussian random field model, i.e. a probabilistic model for spatial data that considers the spatial relationships between data points. The model is then used to predict the labels of unlabeled data points based on the labeled data and with respect to the spatial configuration of all data points. Importantly, they did not limit their inputs to SIFT features but fed entire raw images to an RBF kernel. Nevertheless, because they incorporated both labeled and unlabeled data in their model training process, this approach does not constitute an active, but rather a semi-supervised learning approach.

Semi-supervised learning on image data had garnered significantly more attention than active learning when Gal et al. published their work [6]. In semi-supervised learning, a model is provided with a fixed set of labeled data and a fixed set of unlabeled data. The model can use the unlabeled data to learn about the distribution of the inputs, hoping that this information will aid in learning from the small labeled set as well [4]. Although the learning paradigm is quite different from active learning, this research formed the closest modern literature to active learning of image data at the time. Kingma et al. [11] advanced the field by leveraging deep generative models for semi-supervised learning. They developed a framework that uses variational inference and deep neural networks to model data density, allowing effective generalization from small labeled datasets to large unlabeled ones. Similarly, Weston et al. [12] proposed a method to enhance the generalization ability of neural networks by utilizing nonlinear embedding algorithms. They applied these embeddings as regularizers at the output layer or across multiple layers of deep architectures, which improved both performance and training efficiency. Additionally, Rasmus et al. [13] introduced the Ladder network, combining supervised and unsupervised learning by denoising representations at every layer of the model. This approach achieved state-of-the-art performance on semi-supervised tasks by leveraging hierarchical latent variable models and local learning at each layer.

B. *Review of Contemporary Related Work*

Since the publication of Gal et al.'s paper, there has been significant progress in deep active learning for image classification beyond the literature reviewed at the time.[1] Three active learning strategies have emerged: 1) uncertainty based sampling 2) diversity-based sampling 3) query by committee (ensembles).

Scholars have focused extensively on enhancing uncertainty-based methods akin to Gal et al.'s approach. For instance, in image classification and other computer vision models, MC dropout has been further developed and utilized for uncertainty estimation [16]. As explained in detail in the subsequent sections, MC Dropout involves performing multiple stochastic forward passes through a neural network with dropout applied at prediction time, providing a practical approximation of epistemic model uncertainty in a Bayesian Neural Network (BNN) [17]. This technique has been refined to work better in the typically batch-based acquisition process by increasing sample diversity within acquired batches (BatchBALD) [18]. Still, single-model MC dropout

[1] For a complete overview, confer Ren et al.'s systematic review [4], Zhan et al.'s comparative survey [14], or a benchmarking study specifically conducted for active learning for image classification by Beck et al. [15].

approaches were found to suffer from the mode collapse problem in variational inference which leads to a faulty estimation of the posterior distribution in the BNN [19]. This could be countered by ensembling several MC dropout models but very expensive computationally [20].

In addition, diversity-based approaches have been developed. The most relevant of these is coreset selection, which focuses on selecting a diverse and representative set of data points to improve the generalization capability of the model [21]. Although its authors argue that coreset is generally superior to the method presented throughout this paper, several benchmarking experiments and reviews contradict or question this claim [15], [22], [23]. While effective at capturing the overall structure of the dataset early on, diversity-based approaches are less sensitive to data samples near the decision boundary, which are often more critical for the prediction model [14]. Hybrid approaches also exist: Batch Active learning by Diverse Gradient Embeddings (BADGE) [24] focuses on calculating gradient embeddings for each unlabeled data point by determining the gradient of the loss function with respect to model parameters, assuming each possible class label. It then uses the K-means++ algorithm [25] on these gradient embeddings to ensure the selected batch is diverse and covers a wide input space like in coreset. While found to perform better than coreset [24], BADGE did not outperform uncertainty-based methods in a major evaluation of deep active learning on image classification tasks [15].

Lastly, query-by-committee (ensemble) models have also gained traction: Unlike single-model MC dropout or classic uncertainty-based methods, deep ensembles combine predictions from multiple independent models to capture epistemic uncertainty and increase overall model diversity. They have been found to offer better-calibrated and more reliable uncertainty estimates compared to both single deterministic and Bayesian neural networks; albeit at oftentimes higher computational cost [26]. Beluch et al. [22] demonstrated ensembles' performance specifically in the active learning for image classification setting: Their approach not only outperformed Bayesian methods but coreset, entropy based approaches with pseudo-labeling [27], and methods based on the expected model output change principle [28], too. This is attributed to ensembles effectively counteracting class imbalances during the acquisition process – an issue prevalent in many domains like medical imaging. Although their performance appears promising, ensembles have so far eschewed extensive benchmark studies and scrutiny in the active learning image classification domain [14], [15], [29].

III. Method

A. Convolutional Neural Networks in the Bayesian Setting

This section will focus on presenting the method by Gal et al. [6], where a model was created that can both work with high-dimensional data, i.e. images, and represent prediction uncertainty on this data. The specific class of models employed was the CNN [30]–[32]. Their design allows them to specifically capture spatial and localized information in images, something RBF kernels are not specifically designed for [3]. It is due to this property that they achieved a landmark success in 2012 on ImageNet [33] and continue to dominate the image classification model landscape [29].

As elaborated in I. and II., traditional CNNs lack a quantification of the model's prediction uncertainty and are not naturally suited for active learning settings. For this reason, Gal et al. introduced the concept of a Bayesian Convolutional Neural Network (Bayesian CNN) in 2015 [34] based on their previous work on dropout as Bayesian approximation [17]. In Bayesian statistics, the goal is to update the probability estimate for a hypothesis as additional evidence is provided. For neural networks, this means updating our beliefs about the model parameters (weights) based on the training data. Bayesian CNNs thus follow the same architecture as regular CNNs except that a prior probability distribution is placed over a set of the model's parameters (weights) $\boldsymbol{\omega} = \{W_1, ..., W_L\}$: $\boldsymbol{\omega} \sim p(\boldsymbol{\omega})$. The prior can be a default Gaussian prior with $p(\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\omega}|0, 1)$, equivalent to L2-regularization; or alternatively another prior, such as a Laplace prior, equivalent to L1-regularization [35]. Furthermore, we define a likelihood model

$$p(y = c|\boldsymbol{x}, \boldsymbol{\omega}) = softmax(\boldsymbol{f}^{\boldsymbol{\omega}}(\mathrm{x}))$$

for classification tasks, or a Gaussian likelihood for regression tasks

$$p(y|x, \boldsymbol{\omega}) = \mathcal{N}(y|\boldsymbol{f}^{\boldsymbol{\omega}}(\mathrm{x}), \sigma^2)$$

where $\mathcal{N}(\cdot|\mu, \sigma^2)$ denotes the Gaussian distribution with mean μ and variance σ^2, and $f_{\boldsymbol{\omega}}(x)$ is the output of the CNN given input x and parameters $\boldsymbol{\omega}$.

An important shortcoming of the Bayesian CNN is that exact inference in the model involves computing the exact posterior distribution over the model parameters $\boldsymbol{\omega}$ given the training data $\mathcal{D}_{train}$. Using Bayes' Theorem, we get

$$p(\boldsymbol{\omega}|\mathcal{D}_{train}) = \frac{p(\mathcal{D}_{train}|\boldsymbol{\omega})p(\boldsymbol{\omega})}{p(\mathcal{D}_{train})} \tag{1}$$

(1) can also be stated as posterior $\propto$ likelihood $\times$ prior. The marginal likelihood is given by the integral

$$p(\mathcal{D}_{\text{train}}) = \int p(\mathcal{D}_{\text{train}}|\boldsymbol{\omega})p(\boldsymbol{\omega})\, d\boldsymbol{\omega} \tag{2}$$

which is often analytically intractable to solve because the integral may not have a closed-form solution. Meanwhile, the high dimensionality and complexity of the integral can prohibit numerical integration, too, due to computational cost [1].

To overcome these shortcomings in the Bayesian setting, we have to rely on approximate inference. According to Bishop [1], these can broadly be categorized into stochastic (Markov Chain Monte Carlo) and deterministic approaches (variational inference, expectation propagation). Gal et al.'s approach bridges the two by leveraging dropout, a technique traditionally used for regularization, as a means of performing approximate Bayesian inference [17].

B. Monte Carlo Dropout

Dropout, traditionally used as a regularization technique, involves randomly setting a subset of activations to zero during each training iteration [36]. Mathematically, for a given layer l in the network, if $\boldsymbol{h}_l$ denotes the activations before dropout, and $\boldsymbol{z}_l$ is a binary mask where each element $z_{l,i}$ is drawn independently from a Bernoulli distribution Bernoulli$(1 - p_l)$, the activations after applying dropout are given by:

$$\tilde{\boldsymbol{h}}_l = \boldsymbol{z}_l \odot \boldsymbol{h}_l$$

where $\odot$ denotes element-wise multiplication, and p_l is the probability of an element being dropped. This operation can be seen as sampling from a posterior distribution where each mask configuration corresponds to a different 'thinned' model, thereby approximating the effect of averaging over multiple (non-independent) neural network architectures [3].

Initially proposed as a regularization technique, dropout has been extended to perform variational inference in Bayesian CNNs [17]. When employing dropout during both the training and evaluation (prediction) phases, each forward pass through the network becomes a stochastic forward pass that effectively samples from an approximate posterior distribution. Formally, this process can be conceptualized as variational inference where dropout configures the approximate distribution, $q^*(\boldsymbol{\omega})$, a tractable form within the family of distributions that minimizes the Kullback-Leibler (KL) divergence to the (intractable) true model posterior $p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})$ conditioned on the training data $\mathcal{D}_{\text{train}}$ [34].

Based on Gal and Ghahramani's findings [17], [34], the approximation of the true posterior from (1) by the dropout-induced distribution can be represented as follows in the classification setting with c classes:

$$\begin{aligned} p(y = c|\boldsymbol{x}, \mathcal{D}_{\text{train}}) &= \int p(y = c|\boldsymbol{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}}) d\boldsymbol{\omega} \\ &\approx \int p(y = c|\boldsymbol{x}, \boldsymbol{\omega}) q^*(\boldsymbol{\omega}) d\boldsymbol{\omega} \\ &\approx \frac{1}{T} \sum_{t=1}^{T} p(y = c|\boldsymbol{x}, \boldsymbol{\omega}_t) \end{aligned} \tag{3}$$

where $\boldsymbol{\omega}_t$ represents the weights for the t-th stochastic forward pass under the dropout distribution $q^*(\boldsymbol{\omega})$. Each pass generates a prediction, and the aggregation of these predictions approximates the predictive distribution, capturing the model's epistemic uncertainty effectively. The following section will discuss how specific acquisition functions and suitable approximations thereof based on (3).

C. Acquisition Functions

In our active learning setting, let $\mathcal{M}$ be our model, $\mathcal{D}_{pool}$ our pool of unlabeled data, and $x \in \mathcal{D}_{pool}$ our inputs. An acquisition function $a(x; \mathcal{M})$ is a function of x that our active learning system uses to decide, what samples from $\mathcal{D}_{pool}$ to include in the next training iteration:

$$x^* = \arg\max\nolimits_{x \in \mathcal{D}_{pool}} a(x; \mathcal{M}) \tag{4}$$

Unlike regression, where simple predictive variance is used for the acquisition function, classification tasks present us with more choices in acquisition functions for a in the maximization problem stated in (4) and therefore ways to choose those pool points that are most likely to improve our model.

1) Select those pool points that maximize predictive entropy – denoted as $\mathbb{H}$ [37]:

$$\begin{aligned} &\mathbb{H}[y|\mathbf{x}; \mathcal{D}_{\text{train}}] \\ &:= -\sum_c p(y = c|\mathbf{x}; \mathcal{D}_{\text{train}}) \log p(y = c|\mathbf{x}; \mathcal{D}_{\text{train}}) \end{aligned}$$

2) Pick pool points that are expected to maximize the information gained about the model's parameters. This is equivalent to maximizing the mutual information $\mathbb{I}$ between predictions and model posterior, also known as the Bayesian Active Learning by Disagreement (BALD) [38], and can be stated as follows:

$$\begin{aligned} &\mathbb{I}[y; \boldsymbol{\omega}|\mathbf{x}; \mathcal{D}_{train}] \\ &= \mathbb{H}[y|x; \mathcal{D}_{train}] - \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{train})}[\mathbb{H}[y|\mathbf{x}; \mathcal{D}_{train}]] \end{aligned} \tag{5}$$

where again $\boldsymbol{\omega}$ are our model parameters. Points that maximize $\mathbb{I}[y; \boldsymbol{\omega}|\mathbf{x}; \mathcal{D}_{train}]$ are those where the model is uncertain on average but, at the same time, model parameters that produce disagreeing predictions with high certainty exist. Looking at our neural network architecture, points that maximize the BALD acquisition function result in a high variance on the logits layer of the network, i.e. directly prior to going through the final `softmax` layer.

3) Similar to maximum entropy, we can maximize Variation Ratios to measure model confidence in predictions [39]:

$$\textit{Var-Ratio}[\mathrm{x}] = 1 - \max_y p(y|x, \mathcal{D}_{train})$$

4) Alternatively, we can opt for maximizing the mean standard deviation (STD) ad-hoc [40], [41]

$$\sigma_c = \sqrt{\mathbb{E}_{q(\boldsymbol{\omega})}\left[p(y = c|\mathbf{x}; \boldsymbol{\omega})^2\right] - \mathbb{E}_{q(\boldsymbol{\omega})}\left[p(y = c|\mathbf{x}; \boldsymbol{\omega})\right]^2}$$

$$\overline{\sigma}_{\mathbf{x}} = \frac{1}{C} \sum_c \sigma_c$$

averaged over all c classes $\mathbf{x}$ can take.

5) Finally, as a baseline, we can randomly sample x from $\mathcal{D}_{pool}$, meaning $a(\mathbf{x}) = \text{unif}()$ with unif() being a function that draws from a uniform distribution over the interval $[0; 1]$ which we then use to pick our datapoint.

D. Approximation of Acquisition Functions Using MC Dropout

Actually calculating these acquisition functions requires we specify the posterior, however, this is not an option due to intractability. Hence, we have to approximate the posterior distribution based on our dropout distribution $q^*(\boldsymbol{\omega})$ from (3).

Gal et al. [6] demonstrate this for BALD (5) in the following way:

$$\mathbb{I}[y;\boldsymbol{\omega}|\mathbf{x};\mathcal{D}_{train}]$$

$$= \mathbb{H}[y|x;\mathcal{D}_{train}] - \mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{train})}[\mathbb{H}[y|\mathbf{x};\mathcal{D}_{train}]]$$

Using the definition of entropy and its expectation, for c classes that y can take, $\mathbb{I}[y;\boldsymbol{\omega}|\mathbf{x};\mathcal{D}_{train}]$ is equivalent to

$$-\sum_c p(y=c|\mathbf{x},\mathcal{D}_{\text{train}})\log p(y=c|\mathbf{x},\mathcal{D}_{\text{train}})$$

$$+\mathbb{E}_{p(\omega|\mathcal{D}_{\text{train}})}\left[-\sum_c p(y=c|\mathbf{x},\omega)\log p(y=c|\mathbf{x},\omega)\right]$$

Now, using the identity

$$p(y=c|\mathbf{x};\mathcal{D}_{train}) = \int p(y=c|\mathbf{x};\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathcal{D}_{train})\,d\boldsymbol{\omega}$$

we can reformulate the previous equation and therefore (5) as

$$\mathbb{I}[y;\boldsymbol{\omega}|\mathbf{x};\mathcal{D}_{\text{train}}] = -\sum_c \int p(y=c|\mathbf{x};\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})d\boldsymbol{\omega}$$

$$\cdot\log\int p(y=c|\mathbf{x};\boldsymbol{\omega})p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})d\boldsymbol{\omega}$$

$$+\mathbb{E}_{p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})}\left[\sum_c p(y=c|\mathbf{x};\boldsymbol{\omega})\log p(y=c|\mathbf{x};\boldsymbol{\omega})\right] \quad (6)$$

Finally, we substitute the posterior $p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})$ using our posterior approximation, i.e. the Monte Carlo dropout distribution $q^*(\boldsymbol{\omega})$ from III.B, resulting in

$$\approx -\sum_c \int p(y=c|\mathbf{x};\boldsymbol{\omega})q^*_{\boldsymbol{\theta}}(\boldsymbol{\omega})d\boldsymbol{\omega}$$

$$\cdot\log\int p(y=c|\mathbf{x};\boldsymbol{\omega})q^*_{\boldsymbol{\theta}}(\boldsymbol{\omega})d\boldsymbol{\omega}$$

$$+\ \mathbb{E}_{q^*_{\boldsymbol{\theta}}(\boldsymbol{\omega})}\left[\sum_c p(y=c|\mathbf{x};\boldsymbol{\omega})\log p(y=c|\mathbf{x};\boldsymbol{\omega})\right] \quad (7)$$

$$\approx -\sum_c\left(\frac{1}{T}\sum_t \hat{p}^t_c\right)\log\left(\frac{1}{T}\sum_t \hat{p}^t_c\right)$$

$$+\ \frac{1}{T}\sum_{c,t}\hat{p}^t_c\log\hat{p}^t_c =: \hat{\mathbb{I}}[y;\boldsymbol{\omega}|\mathbf{x};\mathcal{D}_{\text{train}}]$$

The equation defines the approximation as explained in [6], [17]. Note that $\hat{p}^t_c$ is the probability of input $\mathbf{x}$ to take class c with model parameters $\hat{\boldsymbol{\omega}}_t \sim q^*_{\boldsymbol{\theta}}(\boldsymbol{\omega})$:

$$\hat{\mathbf{p}}^t = [\hat{p}^t_1,\ldots,\hat{p}^t_C] = \text{softmax}(f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}))$$

It therefore holds that

$$\hat{\mathbb{I}}[y;\boldsymbol{\omega}|\mathbf{x};\mathcal{D}_{\text{train}}] \xrightarrow{T\to\infty} \mathbb{H}[y|\mathbf{x};q^*_{\boldsymbol{\theta}}] - \mathbb{E}_{q^*_{\boldsymbol{\theta}}(\boldsymbol{\omega})}\left[\mathbb{H}[y|\mathbf{x};\boldsymbol{\omega}]\right]$$

$$\approx \mathbb{I}[y;\boldsymbol{\omega}|\mathbf{x};\mathcal{D}_{\text{train}}]$$

Hence, as the number of samples T approaches infinity, the estimated mutual information $\hat{\mathbb{I}}$ converges to the true mutual information $\mathbb{I}$ given the data and input. All in all, Gal et al.'s approach gives us a computationally tractable estimator of the BALD acquisition function[2] [6].

IV. Experiments

Gal et al. first evaluated and compared various acquisition functions for Bayesian CNNs using the standard MNIST dataset [42] for image classification.[3] They also applied the same active learning process to deterministic CNNs to examine the role of epistemic vs. aleatoric uncertainty. Their approach was then compared to Zhu et al.'s RBF kernel-based model [10], the only real alternative for active learning with high-dimensional data at the time. Due to the lack of comparable models, the authors benchmarked their Bayesian approach against the leading semi-supervised methods available at the time. Finally, Gal et al. demonstrated the capabilities of their approach on the ISIC 2016 melanoma diagnosis dataset [43], highlighting its effectiveness in a real-world setting.

A. CNN Model Setup on MNIST and Comparison of Acquisition Functions

The architecture of the CNN used follows the standard Keras implementation for the MNIST dataset [44]. Specifically, a convolution-ReLU-convolution-ReLU-max pooling-dropout-dense-ReLU-dropout-dense-softmax structure is used, with 32 convolution kernels, a kernel size of 4x4 with 2x2 pooling (for 2D images), a dense (fully connected) layer with 128 units, and dropout probabilities of 0.25 for the first and 0.5 for the second dropout layer. For the training process, the CNN is initially trained on 20 random but balanced data points and a validation set to optimize weight decay comprising 100 data points. 10,000 data points are used as the test set, and the remaining 49,880 data points are kept in the active learning pool at the time of initial training.

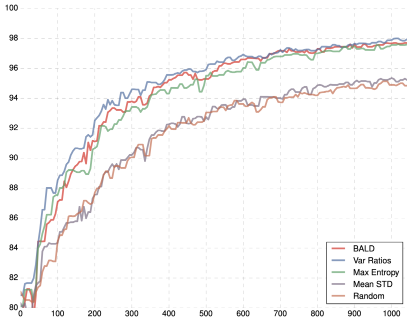

Fig. 1. **Comparison of test error (accuracy) on MNIST vs. number of acquired images from the active learning pool**. Average over three repetitions. Source: [6]

A CNN was set up for each acquisition function studied: BALD, Variation Ratios, Max Entropy, Mean STD, and baseline random acquisition. After the initial training, the model's

[2] This can be applied to any of the aforementioned acquisition functions. An example for approximating variation ratios is provided in the appendix.

[3] MNIST is a standard dataset consisting of 70,000 28x28 pixel black and white images of ten handwritten digits.

test error was determined. Since this was a classification task, test accuracy[4] was used as the metric. Next, leveraging the Bayesian approach by using MC dropout at prediction time with the respective acquisition function, the 10 points that maximized the acquisition function were selected and added to the training set. This process – training, evaluation, acquisition of 10 new data points – was repeated 100 times, resulting in a total of 1,000 data points being added to the initial training set of 20. This was repeated three times for each model, and the results were averaged.

Of the evaluated acquisition functions, BALD, Variation Ratios, and Max Entropy all outperformed Mean STD and the baseline random acquisition function (see Figure 1). The latter two performed roughly similarly. The performance of all acquisition functions can be considered identical, particularly as confidence intervals were not provided. All three functions required significantly fewer images to cross the 10% and 5% test error thresholds on MNIST compared to the baseline random and Mean STD functions.

B. Findings on Model Uncertainty

Gal et al. further evaluate the significance of model uncertainty in Bayesian CNNs by comparing a model with MC dropout to a deterministic CNN using the three best-performing acquisition functions: BALD, Variation Ratios, and Max Entropy (see Figure 2). In deterministic CNNs, dropout is used during training for regularization and deactivated at prediction time. Their experiments demonstrated that Bayesian models, which propagate uncertainty using MC dropout, achieve higher accuracy early on and converge to a higher overall accuracy compared to deterministic models that rely on acquisition functions based on a single (softmax) probability vector output [6]. This superior performance is attributed to the enhanced epistemic uncertainty estimation in the Bayesian model.

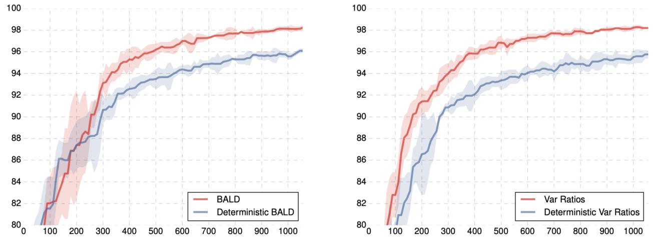

Fig. 2. **Test accuracy vs. number of acquired images for BALD and Var-Ratio acquisition functions.** Bayesian CNN (red) and a deterministic CNN (blue). Source: [6]

C. Bayesian CNNs vs. Available Active and Semi-Supervised Learning Techniques

At the time, the only active learning approach the authors could compare their Bayesian CNNs to was Zhu et al.'s approach using RBF kernels and similarity graphs [10]. However, it was only conceived for binary classification, which required a slight modification of the experiment setup to two-digit classification; everything else remained the same. Just like the CNNs with different acquisition functions (this time leaving out Mean STD), the RBF kernel approach was trained with the same initial training set and then given 10 additional labeled data points from the pool at each iteration. Over 100 iterations, Gal et al. found the RBF kernel approach to significantly underperform compared to all acquisition functions, including the baseline random acquisition function. Results did not improve even after replacing the RBF kernel with a CNN [6].

Subsequently, the performance of Bayesian CNNs with the different acquisition functions was compared to semi-supervised models that have set benchmarks for the MNIST dataset, most notably [11], [13]. It is important to note that in comparison to Gal et al.'s active learning approach, these techniques were trained with 1000 labeled MNIST data points, validation sets of 5000-10000 data points, and the remaining set of 49,000 unlabeled images, whereas the Bayesian CNNs continue to be trained like in the original setup with 20 initial images and 100 iterations à 10 added data points from the pool, thus having access to a mere 1000 labeled training data points in total and not taking into account the distribution of the remaining unlabeled data. However, the validation set used to optimize weight decay (regularization) was extended to comprise 5000 data points instead of the original 100.

TABLE I
TEST ERROR ON MNIST WITH 1000 LABELLED TRAINING SAMPLES, COMPARED TO SEMI-SUPERVISED TECHNIQUES. SOURCE: [6]

Technique	**Test error**
Semi-supervised:	
Semi-sup. Embedding [12]	5.73%
Transductive SVM [12]	5.38%
MTC [45]	3.64%
Pseudo-label [46]	3.46%
AtlasRBF [47]	3.68%
DGN [11]	2.40%
Virtual Adversarial [48]	1.32%
Ladder Network (Γ-model) [13]	1.53%
Ladder Network (full) [13]	0.84%
Active learning with various acquisitions:	
Random	4.66%
BALD	1.80%
Max Entropy	1.74%
Var Ratios	1.64%

In their experiments, the authors found BALD, Max Entropy, and Variation Ratios paired with a Bayesian CNN to outperform several semi-supervised techniques in terms of test error; most notably [11]. An overview of the Bayesian CNNs performance compared to other semi supervised approaches is provided in Table I. After acquiring 1000 training points, the test error for the active learning models was comparable to semi-supervised models, with the Var Ratio acquisition function achieving a 1.64% error rate, slightly higher than the 1.53% error rate of the semi-supervised Γ-model, but without relying on additional unlabeled data.

[4] Accuracy is defined as the proportion of correctly classified instances out of the total number of instances.

D. The Batch Problem

In a subsequent study by the same researchers, it was revealed that batch acquisition methods like those used in Gal et al. often suffer from issues where the samples chosen within a batch tend to be correlated, leading to the selection of redundant information, even with relatively small subsets. This correlation significantly diminishes the overall effectiveness of the process. To tackle this issue, BatchBALD was developed as a solution inspired by diversity-driven approaches [18]. It adapts the Bayesian Active Learning by Disagreement (BALD) acquisition function to preferentially select points that maximize mutual information with the model parameters, thereby minimizing redundancy within the batch. Despite these enhancements geared towards diversity awareness, direct comparisons of BatchBALD with coreset methods have not been explored. Moreover, the initial testing of BatchBALD was confined to less complex datasets like CINIC-10,[5] rather than more challenging datasets like CIFAR-100, where benchmark data for coreset methods are available, such as those discussed by [21].

E. Bayesian CNNs vs. Modern Ensemble Methods

Recent advances in active learning for image classification have shown that ensemble methods now surpass the performance of the Bayesian CNNs proposed by Gal et al. [6] as well as coreset [21] on standard benchmark datasets like MNIST and CIFAR-10. Ensemble-based methods have demonstrated superior accuracy and robustness by leveraging the diversity of multiple learning models initialized with different, random weights. Unlike MC dropout approaches, ensemble methods directly incorporate multiple *independent* model predictions, reducing the variance and improving the confidence of the predictions. At the time of writing, Gal et al. dismissed ensembles as an alternative for the high-dimensional image classification problem due to computational cost. However, advances in computational power now allow for complex ensemble models to be trained efficiently.

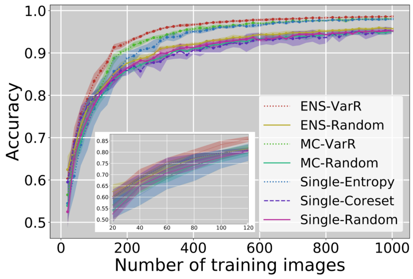

Fig. 3. **Comparison of Keras standard CNN for MNIST test error (accuracy) on MNIST vs. number of acquired images from the active learning pool.** Average and standard deviation (shaded area) over five repetitions. Source: [22]

Beluch et al. [22] took advantage of these enhanced capabilities: They reproduced Gal et al.'s experiments with the standard Keras CNN architecture for MNIST and random acquisition as baseline, albeit only using the, by their interpretation, best-performing acquisition function in Gal et al.: Variation Ratios.[6] In addition to replicating Gal et al.'s setup, they tested an ensemble of 5 deterministic CNNs using the Variation Ratio of the ensemble members' predictions as acquisition function. According to their experiments, on the MNIST dataset, ensemble approaches consistently achieved lower test errors and faster convergence to high accuracy with fewer training images (see Figure 3). The ensemble variation ratio (ENS-VarR) method significantly outperforms Bayesian methods. Unlike Gal et al., Beluch et al. provide the standard deviation based on five repetitions (vs. 3 in Gal et al.) in their results.

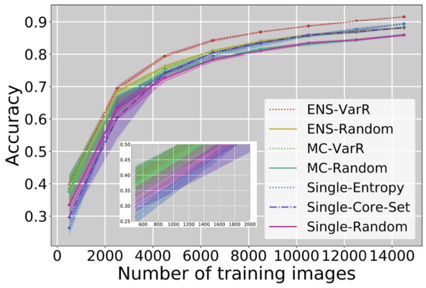

Fig. 4. **Comparison of DenseNet test error (accuracy) on CIFAR-10 vs. number of acquired images from the active learning pool.** Average and standard deviation (shaded area) over five repetitions. Source: [22]

It is important to note that due to the simpler nature of MNIST (28x28 images, black and white, clearly contoured digits) the differences in approaches are less discernible on it. The advantage of CNNs is more pronounced on complex datasets [7] like CIFAR-10 [49] (see Figure 4) or -100. Geometric (coreset) and single approaches proved less effective than Bayesian or ensemble approaches in multiple experiments [22], [53], [54]. This somewhat contradicts the original findings by Sener and Savarese, although the experimental setups differ in the model and amount of training data used [21].

Interestingly, Beluch et al. found that increasing the number of ensemble models did not significantly affect performance; an ensemble with only 3 members outperformed the other active learning approaches, too. Moreover, the computational efficiency of ensemble methods has seen substantial improvements, making them more feasible for large-scale applications compared to computationally intensive Bayesian methods. Techniques like implicit ensembling have made it possible to train these models without the prohibitive computational costs previously associated with ensemble training. These developments underscore a significant shift in active learning strategies, favoring ensemble methods for their practical advantages in handling complex image datasets over Bayesian

[5] A combination of augmented CIFAR-10 [49] and ImageNet [50] images.

[6] BatchBALD was not published yet.

[7] More complex model architectures, namely DenseNet [51] and Resnet18 [52], as well as a larger initial training set were used by Beluch et al. on CIFAR-10 experiments [22], [53].

approaches.

F. Bayesian CNNs and Ensembles in Practice: Melanoma Detection and Diagnosis of Diabetic Retinopathy

In medical imaging (example images provided in Figure 5), active learning is invaluable due to the high cost and scarcity of expert annotations. For melanoma detection, the task involved classifying dermoscopic images of skin lesions as malignant or benign using the 2016 ISIC Archive [43]. In this case, a Bayesian CNN approach was employed, building on the work by Gal et al. with a model based on the VGG16 architecture [55], refined using data augmentation and MC dropout for uncertainty estimation in the sampling process. They started with an imbalanced set of initial training samples. Meanwhile the test was sampled randomly but kept balanced. New samples were iteratively selected based on the BALD acquisition function through 20 MC dropout iterations per unlabeled data point. Interestingly, using Variation Ratios proved unfeasible with this dataset. Overall, the BALD approach significantly improved the model's performance over random acquisition in terms of AUC. The Bayesian approach consistently outperformed the baseline, converging to higher AUC scores faster. This was attributed to BALD avoiding the selection of noisy points with high aleatoric uncertainty but focusing on points that maximize epistemic uncertainty instead [6]. The method was not compared to ensemble methods as they arguably were not developed enough at the time.

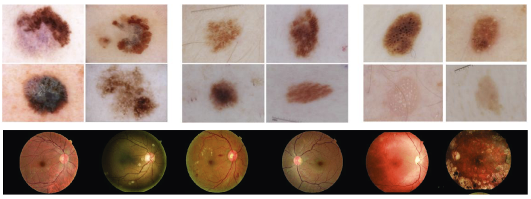

Fig. 5. **Example images for real-world experiments.** Top: Skin cancer (melanoma) lesions (left) vs. benign lesions (center, right). Fundus photographs for the diagnosis of diabetic retinopathy (bottom). Sources: [43], [56].

For diabetic retinopathy – a condition affecting millions globally and leading cause of blindness – active learning has been applied to efficiently use expert-labeled data. Using an InceptionV3 architecture [57] pre-trained on ImageNet [50] and fine-tuned on a dataset of eye fundus images [56], the ensemble-based Variation Ratios (ENS-VarR) method achieved a very high AUC of 0.983 vs. 0.965 for random acquisition after actively selecting 21,000 images out of the total 128,175 [22]. For comparison, a state-of-the-art model with access to the full training dataset (80% more data) achieved an AUC of 0.991 [58]. This demonstrated the capability of ensemble methods on an imbalanced real-world dataset. Unfortunately, the authors did not compare ENS-VarR to MC dropout based VarR in this real-world setting although doing so in the benchmarking studies.

V. Conclusion and Future Research

Deep Bayesian active learning with convolutional neural networks has demonstrated substantial potential in efficiently leveraging limited labeled data for image classification tasks. By integrating MC dropout for approximate Bayesian inference, this approach effectively quantifies epistemic model uncertainty, which is crucial for active learning across various domains. The comprehensive review and experiments presented indicate that acquisition functions such as BALD, Max Entropy, and Variation Ratios, when combined with Bayesian CNNs, outperform traditional active learning methods and even some semi-supervised techniques. This is particularly evident in complex datasets and real-world applications like medical image analysis, where datasets are imbalanced and the cost of obtaining labeled data is high.

However, recent advancements in ensemble methods have shown superior performance compared to single-model Bayesian approaches, particularly with more demanding datasets like CIFAR-10/100. Ensemble techniques leverage the higher diversity of multiple independent models, providing more robust uncertainty estimates and achieving higher accuracy with fewer labeled samples. As computational resources improve, the initial concerns about the computational cost of ensemble methods are diminishing. Nonetheless, further benchmarking of ensemble methods is needed to firmly establish their potential edge over other approaches, as they have hitherto been left out of several benchmarking studies.

Future research should focus on developing more efficient ensemble techniques, exploring hybrid approaches that combine the strengths of Bayesian and ensemble methods, and investigating their applicability in diverse real-world scenarios. Despite these advancements, purely Bayesian approaches will continue to have an advantage when training multiple models is not feasible. This is especially relevant in frontier higher-dimensional domains such as hyperspectral remote sensing images or complex natural language processing, where the computational cost of training multiple models remains a significant consideration.

References

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, United States of America: Springer Nature, 2006.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, ISSN: 1476-4687. DOI: 10.1038/nature14539. [Online]. Available: http://dx.doi.org/10.1038/nature14539.

[3] C. M. Bishop and H. Bishop, *Deep Learning. Foundations and Concepts*. New York, United States of America: Springer Nature, 2024.

[4] P. Ren, Y. Xiao, X. Chang, *et al.*, "A survey of deep active learning," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–40, Oct. 2021, ISSN: 1557-7341. DOI: 10.1145/3472291. [Online]. Available: http://dx.doi.org/10.1145/3472291.

[5] S. Tong, *Active learning: theory and applications*. Stanford University, 2001.

[6] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian active learning with image data," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, Jun. 2017, pp. 1183–1192. [Online]. Available: https://proceedings.mlr.press/v70/gal17a.html.

[7] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, “Multi-class active learning for image classification,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2009. DOI: 10.1109/cvpr.2009.5206627. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2009.5206627.

[8] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, pp. 273–297, 1995.

[9] X. Li and Y. Guo, “Adaptive active learning for image classification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 859–866.

[10] X. Zhu, J. Lafferty, and Z. Ghahramani, “Combining active learning and semi-supervised learning using gaussian fields and harmonic functions,” in *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, vol. 3, 2003, pp. 58–65.

[11] D. P. Kingma, S. Mohamed, D. Jimenez Rezende, and M. Welling, “Semi-supervised learning with deep generative models,” *Advances in neural information processing systems*, vol. 27, 2014.

[12] J. Weston, F. Ratle, and R. Collobert, “Deep learning via semi-supervised embedding,” in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1168–1175.

[13] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, “Semi-supervised learning with ladder networks,” *Advances in neural information processing systems*, vol. 28, 2015.

[14] X. Zhan, H. Liu, Q. Li, and A. B. Chan, “A comparative survey: Benchmarking for pool-based active learning.,” in *IJCAI*, 2021, pp. 4679–4686.

[15] N. Beck, D. Sivasubramanian, A. Dani, G. Ramakrishnan, and R. Iyer, *Effective evaluation of deep active learning on image classification tasks*, 2021. DOI: 10.48550/ARXIV.2106.15324. [Online]. Available: https://arxiv.org/abs/2106.15324.

[16] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?” In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/2650d6089a6d640c5e85b2b88265dc2b-Paper.pdf.

[17] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., ser. Proceedings of Machine Learning Research, vol. 48, New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059. [Online]. Available: https://proceedings.mlr.press/v48/gal16.html.

[18] A. Kirsch, J. van Amersfoort, and Y. Gal, *Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning*, 2019. DOI: 10.48550/ARXIV.1906.08158. [Online]. Available: https://arxiv.org/abs/1906.08158.

[19] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, “Variational inference: A review for statisticians,” *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[20] R. Pop and P. Fulop, “Deep ensemble bayesian active learning: Addressing the mode collapse issue in monte carlo dropout via ensembles,” *arXiv preprint arXiv:1811.03897*, 2018.

[21] O. Sener and S. Savarese, “Active learning for convolutional neural networks: A core-set approach,” *arXiv preprint arXiv:1708.00489*, 2017.

[22] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, “The power of ensembles for active learning in image classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9368–9377.

[23] P. Munjal, N. Hayat, M. Hayat, J. Sourati, and S. Khan, “Towards robust and reproducible active learning using neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 223–232.

[24] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, “Deep batch active learning by diverse, uncertain gradient lower bounds,” *arXiv preprint arXiv:1906.03671*, 2019.

[25] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” Stanford, Tech. Rep., 2006.

[26] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” *Advances in neural information processing systems*, vol. 30, 2017.

[27] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, “Cost-effective active learning for deep image classification,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2016.

[28] C. Käding, E. Rodner, A. Freytag, and J. Denzler, “Active and continuous exploration with deep neural networks and expected model output changes,” *arXiv preprint arXiv:1612.06129*, 2016.

[29] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, “Review of image classification algorithms based on convolutional neural networks,” *Remote Sensing*, vol. 13, no. 22, p. 4712, 2021.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[31] Y. LeCun, B. Boser, J. S. Denker, *et al.*, “Backpropagation applied to handwritten zip code recognition,” *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989. DOI: 10.1162/neco.1989.1.4.541.

[32] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” in *Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed., MIT Press, 1995, p. 3361.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.

[34] Y. Gal and Z. Ghahramani, “Bayesian convolutional neural networks with bernoulli approximate variational inference,” 2016. DOI: 10.48550/ARXIV.1506.02158. [Online]. Available: https://arxiv.org/abs/1506.02158.

[35] T. Hastie, J. Friedman, and R. Tibshirani, *The Elements of Statistical Learning*. Springer New York, 2001, ISBN: 9780387216065. DOI: 10.1007/978-0-387-21606-5. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-21606-5.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[37] C. E. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, pp. 379–423, 1948.

[38] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, *Bayesian active learning for classification and preference learning*, 2011. DOI: 10.48550/ARXIV.1112.5745. [Online]. Available: https://arxiv.org/abs/1112.5745.

[39] L. C. Freeman, *Elementary applied statistics*. New York, United States of America: Wiley, 1965.

[40] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, “Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2016, pp. 680–688. DOI: 10.1109/CVPRW.2016.90.

[41] A. Kendall, V. Badrinarayanan, and R. Cipolla, “Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding,” *CoRR*, vol. abs/1511.02680, 2015. arXiv: 1511.02680. [Online]. Available: http://arxiv.org/abs/1511.02680.

[42] Y. LeCun and C. Cortes, *The mnist database of handwritten digits*, http://yann.lecun.com/exdb/mnist/, 1998.

[43] D. Gutman, N. C. F. Codella, E. Celebi, *et al.*, *Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (isbi) 2016, hosted by the international skin imaging collaboration (isic)*, 2016. arXiv: 1605.01397 [cs.CV].

[44] fchollet, *Keras. experiments: Standard cnn implementation for mnist*. [Online]. Available: https://github.com/fchollet/.

[45] S. Rifai, Y. N. Dauphin, P. Vincent, Y. Bengio, and X. Muller, “The manifold tangent classifier,” *Advances in neural information processing systems*, vol. 24, 2011.

[46] D.-H. Lee *et al.*, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *Workshop on challenges in representation learning, ICML*, Atlanta, vol. 3, 2013, p. 896.

[47] N. Pitelis, C. Russell, and L. Agapito, “Semi-supervised learning using an unsupervised atlas,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2014, pp. 565–580.

[48] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii, “Distributional smoothing with virtual adversarial training,” *arXiv preprint arXiv:1507.00677*, 2015.

[49] A. Krizhevsky, V. Nair, and G. Hinton, “Cifar-10 (canadian institute for advanced research),” 2009. [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html.

[50] ImageNet, *The imagenet data base*, 2021. [Online]. Available: https://www.image-net.org.

[51] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[52] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[53] P. Munjal, N. Hayat, M. Hayat, J. Sourati, and S. Khan, “Towards robust and reproducible active learning using neural networks,” *ArXiv*, vol. abs/2002.09564, 2020.

[54] A. L. Chandra and V. N. Balasubramanian, “Deep active learning toolkit for image classification in pytorch,” *https://github.com/acl21/deep-active-learning-pytorch*, 2021.

[55] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, 2015. arXiv: 1409.1556 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1409.1556.

[56] E. Dugas, J. Jorge, and W. Cukierski, *Diabetic retinopathy detection*, 2015. [Online]. Available: https://kaggle.com/competitions/diabetic-retinopathy-detection.

[57] C. Szegedy, W. Liu, Y. Jia, *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[58] V. Gulshan, L. Peng, M. Coram, *et al.*, “Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs,” *jama*, vol. 316, no. 22, pp. 2402–2410, 2016.

Appendix: Example Approximation of Variation Ratios using MC dropout

To approximate the Variation Ratios in Bayesian CNNs, we can follow the same approach as Gal et al. [6]. Given a training set $\mathcal{D}_{\text{train}}$, input data $\mathbf{x}$, and predicted class y of C classes, the variation ratio is defined as:

$$\textit{Var-Ratio}[\mathbf{x}] = 1 - \max_c p(y = c|\mathbf{x}, \mathcal{D}_{\text{train}})$$

Using the marginal likelihood identity:

$$p(y = c|\mathbf{x}; \mathcal{D}_{\text{train}}) = \int p(y = c|\mathbf{x}; \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})\, d\boldsymbol{\omega}$$

we get:

$$\textit{Var-Ratio}[\mathbf{x}] = 1 - \max_c \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})\, d\boldsymbol{\omega}$$

Next, we can substitute the posterior $p(\boldsymbol{\omega}|\mathcal{D}_{\text{train}})$ with our approximation $q^*(\boldsymbol{\omega})$:

$$\widehat{\textit{Var-Ratio}}[\mathbf{x}] \approx 1 - \max_c \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) q^*(\boldsymbol{\omega})\, d\boldsymbol{\omega}$$

Through T stochastic forward passes with dropout (MC dropout), we get prediction probabilities:

$$\{\hat{p}_c^1, \hat{p}_c^2, \ldots, \hat{p}_c^T\} = \text{softmax}(f^{\hat{\boldsymbol{\omega}}_t}(\mathbf{x}))$$

For each class, average the probability:

$$\hat{p}_c = \frac{1}{T} \sum_{t=1}^{T} \hat{p}_c^t$$

Using:

$$\hat{p}_c = \frac{1}{T} \sum_{t=1}^{T} \hat{p}_c^t \approx \int p(y = c|\mathbf{x}, \boldsymbol{\omega}) q^*(\boldsymbol{\omega})\, d\boldsymbol{\omega}$$

we finally get:

$$\widehat{\textit{Var-Ratio}}[\mathbf{x}] \approx 1 - \max_c \frac{1}{T} \sum_{t=1}^{T} \hat{p}_c^t \approx 1 - \max_c \hat{p}_c$$

Like for BALD, as $T \to \infty$, it holds that:

$$\widehat{\textit{Var-Ratio}}[\mathbf{x}] \approx \textit{Var-Ratio}[\mathbf{x}]$$

The approximated variation ratio converges to the real variation ratio of the model predictions given the input $\mathbf{x}$.