Bayesian Convolutional Neural Networks for Image Classification with Uncertainty Estimation

Fatma Z. Bessai-Mechmache
Research Centre on Scientific and Technical Information, CERIST
Algiers, Algeria
fatmazohrabm@gmail.com

Maya N. Ghaffar
Department of Mathematics and Computer Science, University of Algiers 1
Algiers, Algeria
mayanghaffar@gmail.com

Rayan Y. Laouti
Department of Mathematics and Computer Science, University of Algiers 1
Algiers, Algeria
l.rayaneyasmine@gmail.com

2022 3rd International Conference on Embedded & Distributed Systems (EDiS) | 979-8-3503-9754-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/EDIS57230.2022.9996478

Abstract—**Over the past decade, deep learning has led to a cutting-edge performance in a variety of fields. However, it faces a fundamental constraint which is the treatment of uncertainty. The representation of the model's uncertainty is of significant importance in areas subject to strict safety or reliability requirements. Bayesian deep learning offers a new approach that showcases the degree of reliability of predictions made by neural networks. The present work tests deep learning with Bayesian thinking through a case study of image classification. It puts into practice Bayesian inference to tackle the problem of uncertainty in deep learning and shows its correlation with data quality and model accuracy. To reach this goal, we have implemented a Bayesian convolutional neural network using the variational inference algorithm, Bayes by Backprop. The proposed model was evaluated on an image classification task, with two benchmark datasets. The results' review allowed for validation of the Bayesian approach and showed that it obtains comparable results to those of a non-Bayesian convolutional neural network. Furthermore, the uncertainty of the model was estimated in terms of aleatory and epistemic uncertainty.**

Index Terms—**bayesian deep learning, convolutional neural networks, uncertainty estimation, image classification, bayes by backprop.**

I. Introduction

In recent years, deep learning has led to remarkable success in several fields of artificial intelligence, including computer vision [1], speech recognition [2], and natural language processing [3]. Despite the success of standard deep learning methods in solving various real-world problems, deep neural networks are potentially prone to overfitting and thus require weight regularization measures and similar techniques to reduce it. On the other hand, these networks cannot provide information about the reliability of their predictions. In fact, deep neural networks make point estimates, without taking uncertainty into account [4]. In safety-critical systems, such as self-driving cars and medical diagnoses, being able to tell whether a model is certain of its output can be as important as the accuracy of the prediction. The use of such models often leads to overconfident predictions, and therefore less reliable decisions which can have disastrous consequences.

On May 7, 2016, an assisted driving vehicle collided with a tractor-trailer crossing an uncontrolled intersection on a highway west of Williston, Florida, resulting in the death of the driver. Data obtained from the vehicle indicated that the perception system mistook the white side of the trailer for a bright sky [5]. In a second example, a Google image classification system misidentified two African-Americans as gorillas [6], raising concerns about racial discrimination. Such malfunctions can be avoided if we correctly estimate the uncertainty of the machine learning system.

Over the last few years, Bayesian deep learning (BDL) has emerged as a probabilistic framework that closely integrates deep learning and probability theory to address these problems. BDL offers a solid approach based on the principle of uncertainty that provides an accurate measure of confidence in deep neural networks' predictions.

The present work proposes a Bayesian deep learning approach for image classification using convolutional neural networks (CNNs) based on the Bayes by Backprop algorithm and uncertainty estimation. It also offers a comparison with the standard approach through experiments and results.

This paper is organized as follows: Section II presents related works and the paper's contribution. Section III is devoted to the description of the proposed Bayesian deep learning approach. Section IV gives experimental results and shows the effectiveness of the Bayesian convolutional neural networks. Section V concludes the paper.

II. Related Works

While the number of theoretical contributions to Bayesian deep learning is still growing, the practical applications of Bayesian methods are quite limited. This is mainly due to the lack of efficient algorithms to overcome the computational difficulties imposed by Bayesian inference methods, as well as the lack of understanding of recent contributions to tackle these challenges.

Deodato et al. [7] implemented Bayesian neural networks with variational inference where they estimated the uncertainty of the predicted class labels by calculating a confidence score based on the predictive variance. Their Bayesian approach was tested on MNIST and EMNIST and then applied to a biomedical image dataset. Through their uncertainty analysis, they showed that one can achieve better accuracy by only relying on predictions with high confidence, and identified images as out-of-distribution samples although they shared the same ground-truth labels.

979-8-3503-9754-3/22/$31.00 ©2020 IEEE

Ribeiro et al. [8] proposed a Bayesian self-training methodology for automated data annotation, by providing predictive uncertainty estimates using variational inference and deep neural networks. In their work, they offered ways to minimize the established issue of propagating errors in self-training by including an entropy penalty on the log-likelihood loss to punish overconfident output distributions and facilitate thresholding, and an adaptive sample-wise weight on the influence of predicted pseudo-labeled samples over gradient updates to be inversely proportional to their predictive uncertainty.

Based on Bayesian approaches to deep learning, in this work, we combine recent advances in Bayesian deep learning into a supervised image classification framework. We develop a convolutional neural network using the variational inference algorithm Bayes by Backprop and test the implemented model on two benchmark datasets to validate the proposed approach.

III. Bayesian Deep Learning

Bayesian deep learning is an active area of research that forms a field at the intersection of deep learning and Bayesian probability theory [9]. Bayesian deep learning offers a novel approach based on the idea of uncertainty estimation through probability distributions. Bayesian models extend the power of hierarchical representation of deep learning models by inferring complex posterior probability distributions [10].

A. Bayesian Inference

In a neural network, Bayesian inference calculates the posterior distribution of the parameters given the learning data, $p(\theta|D)$, where θ represents the parameters and D the dataset. The inference process with this approach combines existing information on the model, known as priors, with the data from sampling using Bayes' rule. The results of Bayesian inference are probability distributions known as posteriors.

The general form of Bayes' equation is :

$$p(\theta|D) = \frac{p(D|\theta).p(\theta)}{p(D)} \tag{1}$$

which can also be translated as follows :

$$Posterior = \frac{Likelihood \times Prior}{Evidence} \tag{2}$$

- The prior, $p(\theta)$, is the credibility of the θ values without the data D, whereas the posterior, $p(\theta|D)$, is the credibility of θ values with the data D taken into account.
- The likelihood, $p(D|\theta)$, is the probability that the data could be generated by the model with the parameter value θ.
- The model evidence, $p(D)$, is the overall probability of the data determined by an average of all possible parameter values, weighted by the strength of belief in those parameter values.

The denominator of Bayes' rule, labeled in Equation (2) as the evidence or the model evidence, is also called the marginal likelihood. This term refers to the operation of taking the average of the likelihood, $p(D|\theta)$, across all values of θ, weighted by the prior probability of θ [11].

The computation of posterior distributions is often analytically intractable when the datasets are large. This has led researchers to propose various approximate inference methods to perform these calculations. Two methods predominantly used are:

- **Markov Chain Monte Carlo Methods (MCMC):** These methods use repeated random sampling to approximate the posterior distribution [12]. The main drawback of MCMC methods is that they may, in some cases, take a very long time to converge to the desired distribution [13].
- **Variational Inference:** This method allows avoiding computing the marginal likelihood by directly approximating the posterior distribution with a simpler one [14]. This is done by minimizing the Kullback-Leibler divergence between the two distributions [15].

B. Variational inference

Variational inference [16] is an inference method based on principles of analytical approximation, which allows a more evolutionary estimation of the posterior probability distribution, $p(w|D)$, using a machine learning approach based on optimization. The objective is to approximate the posterior density, $p(w|D)$, by considering a family of candidate densities, $q_\theta(w)$. The approximation is measured using the Kullback-Leibler (KL) divergence between the target and candidate posterior density [12].

The KL divergence is defined as follows :

$$KL(q_\theta(w)||p(w|D)) = \int q_\theta(w) \log \frac{q_\theta(w)}{p(w|D)} dw \tag{3}$$

where θ is the set of variational parameters describing the proposed distribution q of the model's parameters w. Variational Inference minimizes the KL divergence by maximizing the lower bound on the marginal likelihood, also called the Evidence Lower BOund (ELBO), which is defined as follows [12] :

$$\begin{aligned} \log p(D) \quad &\geq ELBO = E_{q_\theta(w)}[\log p(D|w)] \\ &\qquad -KL(q_\theta(w)||p(w)) \end{aligned} \tag{4}$$

where the first term represents the conditional log-likelihood of the data given the parameters, and the second term represents the negative KL divergence.

The KL divergence between the posterior approximation $q_\theta(w)$ and the prior distribution p(w) is not a distance metric, but it can be interpreted as a measure of how far the two distributions are from each other. It is a regularization term because it aims to keep the solution as close as possible to the prior belief defined on the model parameters [7].

C. Bayes by Backprop

In this section, we introduce the basic ideas of the Bayes by Backprop algorithm, which is used in this paper.

Bayes by Backprop [4], is a backpropagation-compatible variational inference method for learning the posterior distribution on the weights of a neural network. This method regularizes the weights by minimizing a compression cost, known as variational free energy.

As noted earlier, variational inference approximates a complex posterior probability distribution, $p(w|D)$, with a simpler approximate or variational distribution, $q_\theta(w)$. The purpose is to get $q_\theta(w)$ to be as close as possible to $p(w|D)$. This is done by searching for the configuration of θ, known as the variational parameter, which minimizes the Kullback-Leibler (KL) divergence between the two distributions [15]. Since the KL divergence is also difficult to calculate precisely, we follow a stochastic variational method [17] [4].

We sample the weights w from the variational distribution $q_\theta(w|D)$, and thus obtain the cost function which is intended to be optimized, i.e. minimized in relation to θ, during training [4]:

$$\begin{aligned} F(D,\theta) \approx \sum_{i=1}^{n} & log q_\theta(w^{(i)}|D) \\ & - log p(w^{(i)}) \\ & - log p(D|w^{(i)}) \end{aligned} \tag{5}$$

where n is the number of examples and $w^{(i)}$ is the ith sample from the variational posterior $q_\theta(w^{(i)}|D)$.

D. Predictive Uncertainty

Uncertainty is usually associated with various concepts such as unpredictability, imprecision, and variability. In Bayesian modeling, there are two main types of uncertainty one can model, which are aleatory uncertainty and epistemic uncertainty [18].

Aleatory uncertainty is due to the natural variability inherent in the data [18]. Noise and lack of information are some of the many factors that increase aleatory uncertainty in our data.

On the other hand, epistemic uncertainty captures the learning model's lack of knowledge due to limited data. Therefore, epistemic uncertainty can be reduced by collecting more data or by improving the learning model based on a better understanding of the entities represented.

Aleatory uncertainty, in contrast, cannot be reduced with the availability of more data that will only serve to provide a better representation of this type of uncertainty [19].

In classification tasks, what matters is the predictive probability distribution $p_D(y^*|x^*)$, where x^* is a new sample data and y^* is its predicted class. As we mentioned earlier, this probability distribution is often intractable and must be approximated. To approximate this integral, we use Monte Carlo samples from the approximating distribution q [20]:

$$\begin{aligned} E_q[p_D(y^*|x^*)] &= \int q_0(w|D) p_w(y|x) dw \\ &\approx \frac{1}{T}\sum_{t=1}^{T} p_{w_t}(y^*|x^*) \end{aligned} \tag{6}$$

where T is the predefined number of samples. This estimator allows us to measure the uncertainty of our predictions by defining the predictive variance. This measure can be divided into aleatory and epistemic uncertainty, as follows [20]:

$$\begin{aligned} Var_q(p(y^*|x^*)) &= E_q[yy^T] - E_q[y]E_q[y]^T \\ &= \frac{1}{T}\sum_{t=1}^{T} diag(\hat{p}_t) - \hat{p}_t\hat{p}_t^T \\ & \frac{1}{T}\sum_{t=1}^{T} (\hat{p}_t - \bar{p})(\hat{p}_t - \bar{p})^T \end{aligned} \tag{7}$$

The first term, $\frac{1}{T}\sum_{t=1}^{T} diag(\hat{p}_t) - \hat{p}_t\hat{p}_t^T$, represents the aleatory uncertainty while the second term, $\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_t - \bar{p})(\hat{p}_t - \bar{p})^T$, represents the epistemic uncertainty.

IV. Experimental Results

In this section, we present the experimental results by demonstrating the performance of the Bayesian CNN on both the MNIST and CIFAR-10 datasets with two neural network architectures.

A. Experimental Setup

This work was done using Python libraries. As for the deep learning aspect, we mainly used PyTorch which is a high-level framework that allows us to easily manipulate learning layers.

For the conducted experiments, we implemented a Bayesian image classification CNN model, as well as a non-Bayesian model, in order to compare the results obtained.

As outlined in the following sections, performance analysis of the Bayesian and non-Bayesian models is carried out by evaluating both of the CNN architectures (LeNet-5 and AlexNet) on two well-known image classification benchmark datasets, MNIST and CIFAR-10.

B. Datasets

In this section, we present the datasets used in the experiments.

1) MNIST: The MNIST (Mixed National Institute of Standards and Technology) database [22] is a large dataset that contains 70,000 examples of handwritten digits which was introduced by Yann LeCun in 1998.

2) CIFAR-10: The CIFAR-10 database was introduced in 2009 by the Canadian Institute for Advanced Research (CIFAR), in which it was first used [21].

The database consists of 60,000 color images (RGB) representing 10 different types of objects, namely, an airplane, a car, a bird, a cat, a deer, a dog, a frog, a horse, a boat, and a truck.

C. Neural Network Architectures

We considered LeNet-5 [22] and AlexNet [23] as the base architectures for our neural networks. The architectures used for the experiments are illustrated in Table I and II. We used the same architectures for both the standard and Bayesian models in order to fairly compare both models' performances.

Layer type	Width	Stride	Padding	Input shape
Convolution (5 * 5)	6	1	0	1*32*32
Max-pooling (2 * 2)	-	2	0	6*28*28
Convolution (5 * 5)	16	1	0	1*14*14
Max-pooling (2 * 2)	-	2	0	16*10*10
Fully-connected	120	-	-	400
Fully-connected	84	-	-	120
Fully-connected	10	-	-	84

TABLE I: LeNet-5 architecture [24]

Layer type	Width	Stride	Padding	Input Shape
Convolution (11 * 11)	64	4	5	3*32*32
Max-pooling (2 * 2)	-	2	0	64*32*32
Convolution (5 * 5)	192	1	2	64*15*15
Max-pooling (2 * 2)	-	2	0	192*15*15
Convolution (3 * 3)	384	1	0	192*7*7
Convolution (3 * 3)	256	1	0	384*7*7
Convolution (3 * 3)	128	1	0	256*7*7
Max-pooling (2 * 2)	-	2	0	128*7* 7
Fully-connected	128	-	-	128

TABLE II: AlexNet architecture [21].

D. *Classification Accuracy*

The training, validation, and test accuracy of the trained Bayesian and non-Bayesian models on the MNIST and CIFAR-10 datasets are displayed in Table 4 below. The accuracy, here, represents the percentage of correctly predicted instances from the total number of instances.

The results displayed in Table 3 show that the Bayesian models achieved comparable accuracy results to the standard models. We can see that the standard AlexNet model is overfitting on the CIFAR-10 dataset, judging by the large gap between the training accuracy and the validation accuracy, whereas the Bayesian AlexNet model isn't.

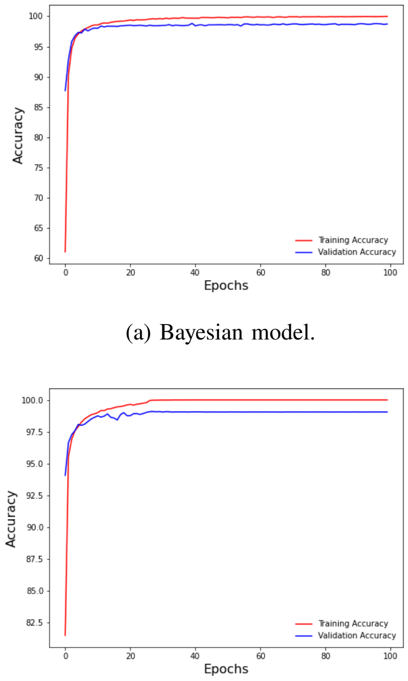
The slight difference between the training accuracy and the validation accuracy of the Bayesian AlexNet model shows the reliability of the Bayesian method which integrates natural regularization effects through inference.

On the other hand, it is clear that the CIFAR-10 dataset has obtained significantly lower accuracy results than the MNIST dataset, regardless of the chosen network architecture. We believe that this is due to the more complex nature of CIFAR-10 images.

Model	Dataset	Training	Validation	Test
Bayesian LeNet-5	MNIST	99.97%	98.71%	99.04%
	CIFAR-10	70.81%	61.74%	60.63%
Non-Bayesian LeNet-5	MNIST	99.96%	99.13%	99.15%
	CIFAR-10	73.87%	65.07%	65.93%
Bayesian AlexNet	MNIST	99.91%	98.89%	98.93%
	CIFAR-10	65.57%	60.37%	62.66%
Non-Bayesian AlexNet	MNIST	99.96%	98.99%	99.15%
	CIFAR-10	95.04%	65.59%	64.82%

TABLE III: Comparison of the training, validation, and test accuracy of the trained Bayesian and non-Bayesian models on the MNIST and CIFAR-10 datasets.

Figure 1 shows the training and validation accuracy of the Bayesian and the standard LeNet-5 models. One thing to observe is that in the first iterations, the Bayesian models start with a slightly lower validation accuracy compared to the same non-Bayesian models. However, as the iterations progress, the results quickly get closer and are almost identical at the end of the 100 learning epochs.



(a) Bayesian model.

(b) Non-Bayesian model.

Fig. 1: Training and validation accuracy of the trained Bayesian and non-Bayesian LeNet-5 models on the MNIST dataset.

E. Training Time

Table IV below compares the training time of the Bayesian and non-Bayesian AlexNet models on the MNIST dataset, depending on the type of processor used.

We can see that training while using a CPU is significantly more time-consuming than using a GPU. This is due to the complexity of the deep Bayesian network, which requires the use of a GPU to run smoothly. However, even while using a GPU the training time is already expensive for a small dataset like MNIST. Compared to the non-Bayesian model with the same architecture, the training time of the Bayesian model with the GPU is almost three times as long, and almost four times as long without the GPU.

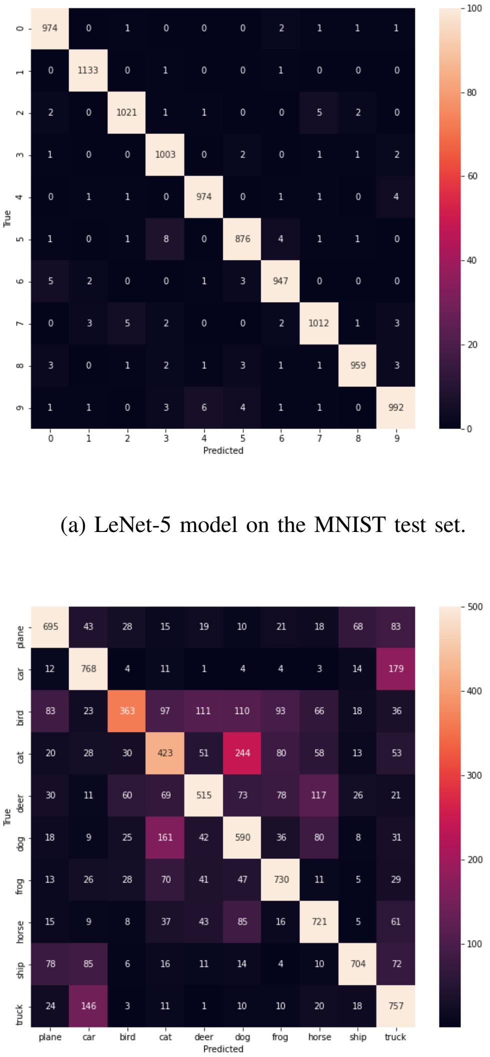
Processor	**Model**	**Training time**
CPU	Bayesian AlexNet	5 hours, 5 minutes and 43 seconds
	Non-Bayesian AlexNet	2 hours, 53 minutes and 12 seconds
GPU	Bayesian AlexNet	1 hour, 5 minutes and 36 seconds
	Non-Bayesian AlexNet	22 minutes and 53 seconds

TABLE IV: Learning time of the Bayesian and non-Bayesian AlexNet models on the MNIST dataset.

F. Confusion Matrices

Figure 2 (a) and (b) shows the confusion matrix for the Bayesian LeNet-5 model on the MNIST test set and the confusion matrix for the Bayesian AlexNet model on the CIFAR-10 test set, respectively. Each row of the matrix represents the instances of a real class while each column represents the instances of a predicted class.

As seen in Figure 2, the largest values are centralized on the diagonal of the matrices, thus representing correct predictions. However, we notice a slight confusion between some classes. In the case of MNIST, certain digits are more difficult to distinguish, such as ”5” with ”3”, ”9” with ”4”, ”6” with ”0”, or ”7” with ”2”. In the case of CIFAR-10, certain classes such as ”car” and ”truck”, ”cat” and ”dog”, or ”deer” and ”horse”, may be confused when making predictions.



(a) LeNet-5 model on the MNIST test set.

(b) AlextNet model on the CIFAR-10 test set.

Fig. 2: Confusion matrices for the Bayesian models.

G. Uncertainty Estimation

For each instance, we calculated the predictive variance, which is a measure of confidence reflecting the uncertainty of the predictions. The aleatory and epistemic uncertainty modelling for a single test image for each dataset is presented in Figure 3 and 4.

In the histograms below, each bar represents one of the predictable classes. The higher bars represent the classes that the model considers closest to the predicted class. In the two cases presented, the number ”3” can be misinterpreted as being a ”5”, and similarly, the object ”cat” can be confused with the object ”dog”. This is because these classes can be seen as confusing or ambiguous observations.

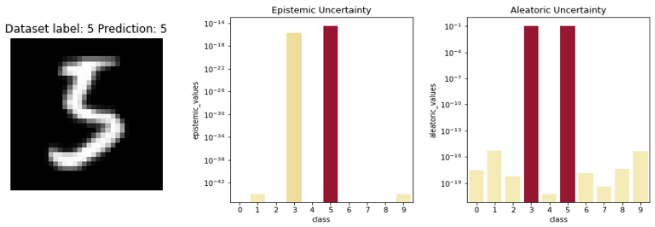


Fig. 3: Aleatory and epistemic uncertainty of a test image from the MNIST dataset.

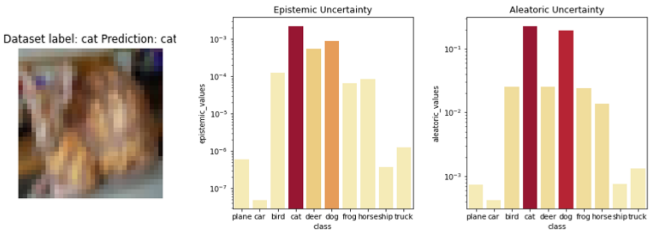


Fig. 4: Aleatory and epistemic uncertainty of a test image from the CIFAR-10 dataset.

We report, in Table V, the average of the aleatory and epistemic uncertainty for the Bayesian LeNet-5 model.

Dataset	**Aleatory uncertainty**	**Epistemic uncertainty**	**Validation accuracy**
MNIST	0.0006	1.288e-15	98.71%
CIFAR-10	0.0427	2.321e-15	61.74%

TABLE V: Aleatory and epistemic uncertainty on the MNIST and CIFAR-10 datasets.

CIFAR-10's aleatory uncertainty is considerably greater than MNIST's. Since aleatory uncertainty measures the irreducible variability of the data, the poorer quality of the CIFAR-10 images may have resulted in higher aleatory uncertainty. The epistemic uncertainty of CIFAR-10, on the other hand, is nearly three times greater than MNIST's, which is justified since the epistemic uncertainty decreases as the validation accuracy of the model increases.

V. Conclusion and Future Work

In this work, we presented a Bayesian deep learning approach for image classification based on a convolutional neural network, using the Bayes by Backprop algorithm as a Bayesian inference method.

The proposed approach has shown good performance results in terms of learning accuracy while avoiding overfitting. Moreover, it allowed us to estimate both aleatory and epistemic uncertainty of the learning model's predictions.

Future work will extend the experimental study to larger datasets and more advanced CNN architectures. We also intend to apply this approach to other tasks in which uncertainty estimation can be even more crucial, such as object detection and semantic segmentation.

References

[1] Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience, 2018, pp. 130–143.

[2] Nassif, A., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech Recognition Using Deep Neural Networks: A Systematic Review. IEEE Access, 7(1109), pp. 19143–19165.

[3] Torfi, A., Shirvani, R., Keneshloo, Y., Tavvaf, N., & Fox, E. (2020). Natural Language Processing Advancements By Deep Learning: A Survey. ArXiv, abs/2003.01200.

[4] Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight Uncertainty in Neural Networks. In Proceedings of the 32nd International Conference on Machine Learning (ICML 2015).

[5] Smith, S. (2017). NTSB: Fatal Crash Involving Tesla Autopilot Resulted from Driver Errors, Overreliance on Automation.

[6] Guynn, J.(2015). Google Photos labeled black people 'gorillas'. https://www.usatoday.com/story/tech/2015 /07/01/google-apologizes-after-photos-identify-black-people-as-gorillas/29567465/.

[7] Deodato, G., Ball, C., & Zhang, X. (2019). Bayesian Neural Networks for Cellular Image Classification and Uncertainty Analysis. Preprint.

[8] Ribeiro, F., Caliva, F., Swainson, M., Gudmundsson, K., Leontidis, G., & Kollias, S. (2019). Deep Bayesian Self-Training. Neural Computing and Applications, 32, pp. 4275–4291.

[9] Kendall, A. (2017). Deep Learning Is Not Good Enough, We Need Bayesian Deep Learning for Safe AI.

[10] McAllister, R., Gal, Y., Kendall, A., Wilk, M. V., Shah, A., Cipolla, R., & Weller, A.(2017). Concrete Problems for Autonomous Vehicle Safety: Advantages of Bayesian Deep Learning. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-17).

[11] Kruschke, J. (2015). Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan. Academic Press.

[12] Sambasivan, R., Das, S., & Sahu, S. (2020). A Bayesian Perspective of Statistical Machine Learning for Big Data. Comput. Stat., 32, pp. 893–930.

[13] Hinton, G., & Neal, R. (1995). Bayesian Learning for Neural Networks. Ph.D. thesis.

[14] Jordan, M., Ghahramani, Z., Jaakkola, T., & Saul, L. (1999). An Introduction to Variational Methods for Graphical Models. Machine Learning, 37, pp. 183–233.

[15] Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In Proceedings of The 33rd International Conference on Machine Learning, Vol.48, pp. 1050–1059.

[16] Hinton, G., & Camp, D. (1993). Keeping the Neural Networks Simple by Minimizing the Description Length of the Weights. In Proceedings of the Sixth Annual Conference on Computational Learning Theory, pp. 5–13.

[17] Graves, A.(2011). Practical Variational Inference for Neural Networks. In Proceedings of the 24th International Conference on Neural Information Processing Systems, pp. 2348–2356.

[18] Kiureghian, A., & Ditlevsen, O. (2009). Structural Safety: Aleatory or Epistemic? Does It Matter? Structural Safety, 31, pp. 105–112.

[19] Kendall, A., & Gal, Y. (2017). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 5580–5590.

[20] Kwon, Y., Won, J., Kim, B., & Paik, M. (2018). Uncertainty Quantification Using Bayesian Neural Networks in Classification: Application to Ischemic Stroke Lesion Segmentation. In Proceedings of the 1st Conference on Medical Imaging with Deep Learning (MIDL2018).

[21] Krizhevsky, A. (2009). Learning Multiple Layers of Features from Tiny Images. Tech. rep., University of Toronto.

[22] LeCun, Y., Cortes, C., & Burges, C. (1998b). MNIST Handwritten Digit Database. http://yann.lecun.com/exdb/mnist/.

[23] Krizhevsky, A., Sutskever, I., & Hinton, G.(2012). ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Vol. 1, pp. 1097–1105.

[24] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998a). Gradient-Based Learning Applied to Document Recognition. In Proceedings of the IEEE, Vol. 86, pp. 2278–2324.