Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it
Bonn-Aachen
International Center for
Information Technology

R&D Project Proposal

# Uncertainty Estimation for Quantized Deep Learning Models: Comparative Study

*Mohan Raj Nadarajan*

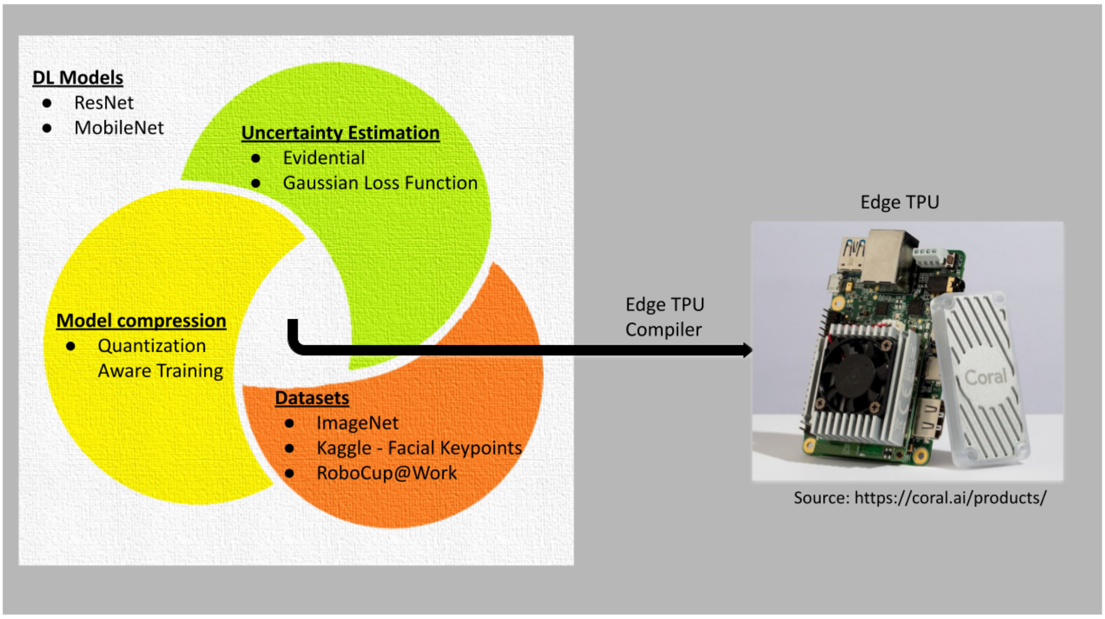
Supervised by

Prof. Dr. Sebastian Houben
M.Sc. Deebul Sivarajan Nair

December 2022

# 1 Introduction

The effectiveness of image processing deep learning models over other conventional machine learning models is indisputable. Though the deep learning models exhibits higher accuracy, the wrong predictions are with higher value of confidence, which will not only fail the task-but also put human lives at jeopardy in the case of safety critical and real world applications[8]. As a result, it is very important for the deep learning models to be aware of confidence in its prediction, alternatively the uncertainty in its prediction .



Source: https://coral.ai/products/

Figure 1: Project components

The predictive uncertainty in a deep learning model is of two types: Aleatoric uncertainty and Epistemic uncertainty or a combination of both[20]. Aleatoric uncertainty also known as data uncertainty is due to the noise in the training data. Epistemic uncertainty also known as model uncertainty is due to uncertainty in the parameters of the model. The epistemic type of uncertainty can be reduced by adding more input data to the model. The deep learning community has provided different uncertainty estimation methods like Monte Carlo dropout[7], stochastic batch normalization[2], test-time data augmentation[3], deep ensembles[16], selective classification[9], deterministic[26] and evidential[24] approaches.

The state-of-the-art uncertainty estimation methods are mainly proposed for GPU based inference and there does not exist much research on uncertainty estimation in deep learning models for edge devices. Edge AI combines artificial intelligence and edge computing. The advancement in edge devices for AI applications to make real-time insights along with improved privacy, reduced latency and high availability are revolutionizing the world's largest industries and expanding business outcomes across all sectors. Edge AI runs on a wide range of hardware from micro-controller to tensor processing devices for real world applications like smart watches, autonomous cars, mobile phones.

Training or inferencing the neural network algorithms on general purpose hardware like CPU (von Neumann architecture) or GPU is inefficient due to higher number of multiply-accumulate operations[12]. One way to accelerate neural network workload is using the specialized matrix processor called Tensor Processing Unit. They are of domain specific hardware architecture for deep learning and is a combination of matrix multiplication unit (MXU), vector processing unit and high-bandwidth memory. The MXU contains series of multiply-accumulators, connected directly to each other forming a systolic array architecture. This R&D works evaluates the uncertainty estimation for deep learning models deployed on the Device Under Test(DUT) such as Google Coral - USB accelerator and OAK-D, driven by Intel Movidius Myriad X VPU.

Most of the edge TPUs supports only network compressed deep learning model, in order to accelerate the performance of neural network workloads. The popular compression technologies are quantization, pruning and knowledge distillation. Quantization is the process of lowering compute demand by converting 32-bit representation of parameter data into lower representations like 8-bit, 16-bit or others to make the model more compatible with edge device hardware architecture. This helps to achieve a reduced memory footprint, significantly reducing the number of transistors required in the chip. The three types of quantization methods are Quantization-aware Training(QAT), Post-Training Static Quantization(PTQ)

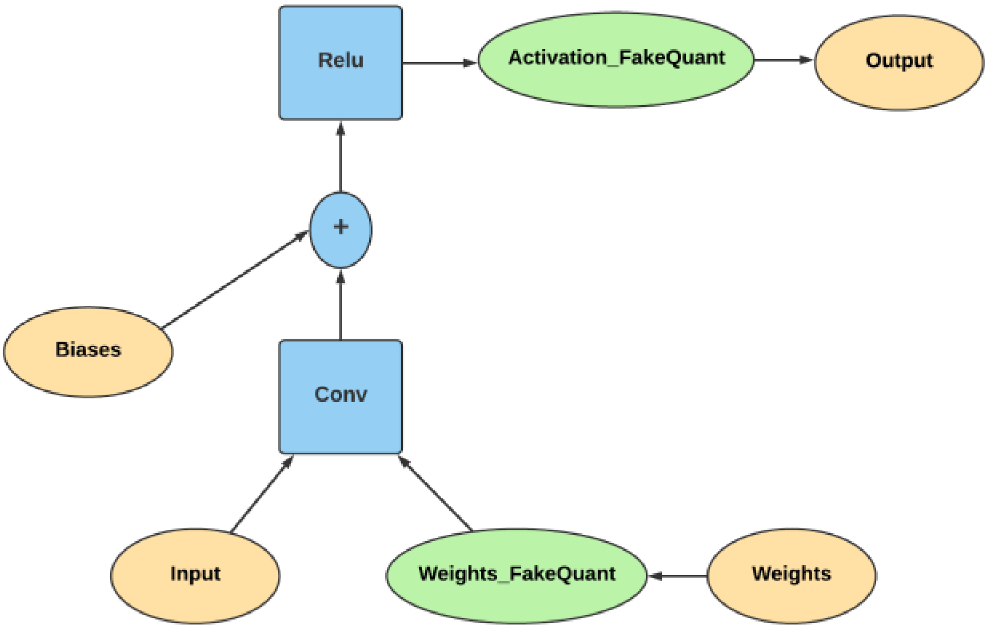


Figure 2: Quantization-aware Training. Duplicated from [5]

and Post-Training Dynamic Quantization[5]. QAT uses ”fake quantized” nodes in forward & backward passes to simulate the impact of lower bit representation during training and is more accurate than post training quantization methods.

The deep learning based image processing tasks using edge devices is widely used in robotics, health care and manufacturing machine applications like quality inspection[25], metrology and packaging. However, there does not exist much work on uncertainty estimates for quantized deep learning models. This R&D project investigates the impact of quantization with different uncertainty estimation methods and also deploy them on edge devices.

**Why is it important?**

Almost every industry is trying to boost automation for enhancing workflow, productivity, and safety with a range of tasks in unstructured world using edge AI. The deep learning model based inference engines deployed on edge devices makes deci-

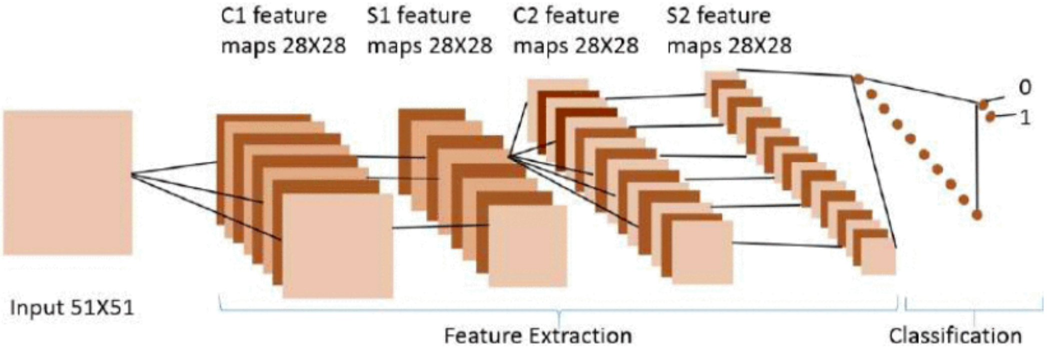


Figure 3: Deep neural network. Duplicated from [14]

sion for real-world problems. So it is important to study the uncertainty estimates for quantized model. This helps in the risk assessment for the decisions to be made for the model's prediction. The incorrect predictions of the model in safety-critical applications can endanger human lives. For instance, IBM's supercomputer Watson recommended 'incorrect and unsafe' cancer treatments[22].

One of the applications, uses a deep learning model to classify objects in the KUKA youBot[4], an omnidirectional mobile manipulator for education and research. The neural network workload is deployed on the robot controller hardware directly and due to its high computing demands, active perception is not possible. This robot is used in RoboCup@Work league[15] and one of the tasks is to pick a desired object from the multiple station environment. In one of the scenarios, the robot accidentally picked up the wrong object, due to wrong predictions made with high confidence. However, if the uncertainty score is calculated along with the model's prediction, a recovery action will be performed. And deploying this neural network workload in an edge device can help to perform active perception.

As a result, it is important to evaluate differences in uncertainty estimates between quantized and non-quantized deep learning models. This study performs qualitative analysis of deep learning models deployed on edge devices, accelerating the growth of safety critical edge AI applications.

## 1.1 Problem Statement

The rise in the popularity of edge computing throws new demands to computing platforms with regards to performance and energy efficiency. In safety critical and real world applications[19], the guarantees for decision making while using quantized deep learning model is vital and is required to evaluate the model uncertainty. The model must be aware of the fact that it will ultimately be quantized, so that it can perform all weight adjustments accordingly during training, to yield higher accurate predictions. QAT outperforms post training quantization methods and is used in this research.

This R&D aims to find and compare the probability distribution functions $f_n$, uncertainty in model predictions, where n $\in \mathbb{N}$, representing different uncertainty estimation methods. The function composes input data x and either non-quantized model parameters $\theta$ or quantized model parameters $\theta'$. In order to evaluate the impact of quantization from the perspective of prediction reliability, the probability distribution functions $f_1(x, \theta)$ and $f_1(x, \theta')$ are compared. In addition, this project compares the functions $f_1(x, \theta')$, $f_2(x, \theta')$, $f_3(x, \theta')$.......$f_n(x, \theta')$ to evaluate the uncertainty estimation methods for the quantized deep learning models.

## 1.2 Research Questions

What are the differences in uncertainty estimates of quantized deep learning models with different uncertainty estimation methods for classification, regression and segmentation tasks?

- Taxonomy of uncertainty estimation methods in deep learning with special focus on single pass uncertainty estimation method
- Literature search on different quantization methods and which are supported by the DUT
- How does evidential loss function impact the QAT?

## 2 Related Work

MLPerf[21], coalition of artificial intelligence accelerators from industry, academia and research labs, aiming to provide state-of-the-art performance evaluation for training and inference tasks. The internal structure of the MLPerf inference submission system has four components and are system under test(SUT), dataset, load generator(LoadGen), and an accuracy script. The LoadGen is a configuration file which handles the traffic generation, loading data for inference and measuring performance. The query format created by LoadGen for different scenarios are single stream, multistream, server and offline. In single stream format, the next query is sent only after the completion of previous query. In multi stream format, the set of inferences per query is sent periodically with a predefined time interval. In server format, the query is random and the SUT responds back within a defined latency limit. In offline format, the whole test data is sent as a batch and latency is not constrained. An empirical study to evaluate the system is performed by MLPerf with ResNet-50 v1.5 and MobileNet-v1 224 models on ImageNet dataset for classification task. The important contribution of this work is to identify the metrics and inference scenarios where AI accelerators are most useful.

The most common benchmarking metrics of deep learning hardware accelerators are energy efficiency, performance and power[17] with their measurement terms: Operations per Watt, Time per inference, and Watt respectively. The aim of Domain Specific Architectures (DSA) is to accelerate inference related operations with reasonable power budget and the success or failure of these DSAs are determined by these benchmarking metrics in the state-of-the-art evaluations.

Achterhold et al. created a complex pruning and quantization strategy for pointwise NNs, but with the help of Bayesian inference[1]. The author trained a Bayesian neural network which is pruning & quantisation friendly and with improper priors. They later converted it to pointwise NNs to achieve reduced memory, but the final non quantized Bayesian NNs were not able to estimate uncertainty because of improper priors[6]. This R&D work is to learn a quantized NN directly and predicting model uncertainty considering a range of uncertainty estimation methods without changes in model architecture.

# 3 Project Plan

## 3.1 Work Packages

The R&D project contains the following work packages

WP1 **Literature search**

- Taxonomy of neural network compression methods for deep learning model
- Literature search on quantization aware training for deep learning model
- Literature search on estimating uncertainty for deep learning models
- Literature search on uncertainty estimation evaluation methods

WP2 **Experimental setup and analysis**

- Train ResNet[10] DNN model with single pass uncertainty estimation methods for classification task on ImageNet[23] dataset
- Perform experiments to estimate ResNet model uncertainty for classification task on GPU
- Train ResNet DNN model with quantization technique and single pass uncertainty estimation methods for classification task on ImageNet dataset
- Perform experiments to estimate quantized ResNet model uncertainty for classification task on GPU
- Comparative evaluation of the model uncertainty estimates for classification task between quantized and non-quantized deep learning models

WP3 **Mid term report**

- Compile quantized ResNet model to support deploying on Google Coral - USB Accelerator and OAK-D with their respective compilers

- Perform experiments to estimate quantized ResNet model uncertainty for classification task on Google Coral - USB Accelerator and OAK-D devices
- Train MobileNet[11] DNN model with quantization technique and single pass uncertainty estimation methods for classification task on ImageNet dataset
- Compile quantized MobileNet model to support deploying on Google Coral - USB Accelerator and OAK-D with their respective compilers
- Perform experiments to estimate quantized MobileNet model uncertainty for classification task on Google Coral - USB Accelerator and OAK-D devices
- Comparative evaluation of the model uncertainty estimates for difference in deep learning models, single pass uncertainty estimation methods and edge devices

WP4 **Regression Task**

- Train ResNet DNN model with quantization technique and single pass uncertainty estimation methods on Facial Keypoints dataset
- Compile quantized ResNet model to support deploying on Google Coral - USB Accelerator and OAK-D with their respective compilers
- Perform experiments to estimate quantized ResNet model uncertainty for regression task on Google Coral - USB Accelerator and OAK-D devices
- Train MobileNet DNN model with quantization technique and single pass uncertainty estimation methods on Facial Keypoints dataset[13]
- Compile quantized MobileNet model to support deploying on Google Coral - USB Accelerator and OAK-D with their respective compilers
- Perform experiments to estimate quantized MobileNet model uncertainty for regression task on Google Coral - USB Accelerator and OAK-D devices

- Comparative evaluation of the model uncertainty estimates for difference in deep learning models, single pass uncertainty estimation methods and edge devices

WP5 **Segmentation Task**

- Train ResNet DNN model with quantization technique and a single pass uncertainty estimation method on RoboCup@Work dataset [18]
- Compile quantized ResNet model to support deploying on Google Coral - USB Accelerator
- Perform experiments to estimate quantized ResNet model uncertainty for segmentation task on Google Coral - USB Accelerator device

WP6 **Project report**

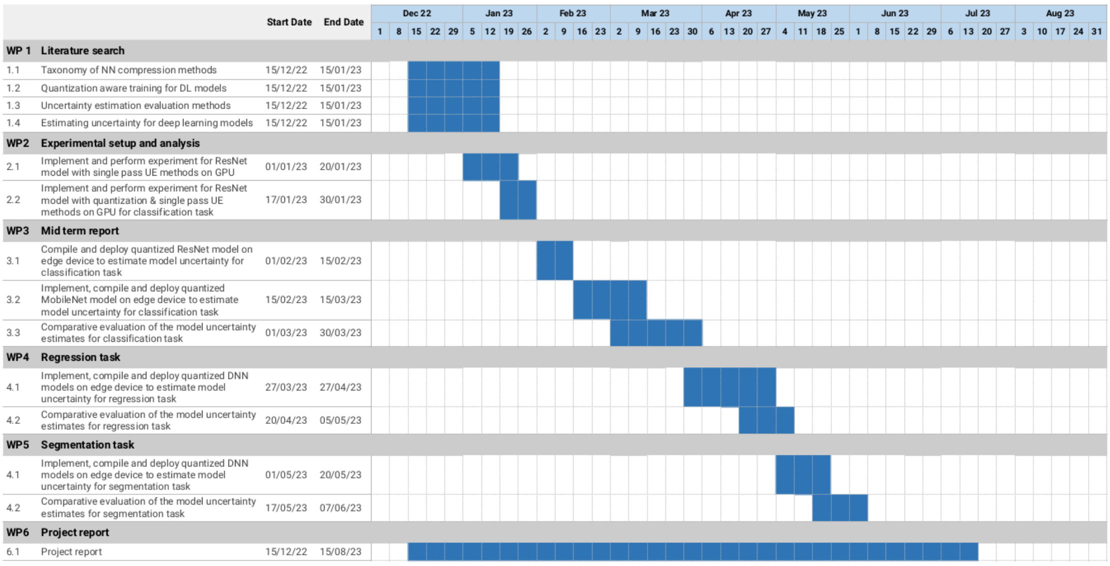
- Documentation of neural network compression techniques taxonomy for deep learning models
- Documentation of state-of-the-art uncertainty estimation methods for classification task
- Documentation of state-of-the-art uncertainty estimation evaluation methods
- Documentation of the quantization techniques support in the edge devices under test
- Documentation of uncertainty estimates for ResNet model with difference in tasks, dataset and uncertainty estimation methods
- Documentation of uncertainty estimates for MobileNet model with difference in tasks, dataset and uncertainty estimation methods
- Documentation of conclusions and recommendations for future work
- Draft R&D report explaining research findings
- Final R&D report explaining research findings

## 3.2 Milestones

| M | Milestone | Checkpoints |
|---|---|---|
| 1 | Literature search | Taxonomy of NN compression methods. Read top research papers on uncertainty estimation for DL models |
| 2 | Experimental setup | Train ResNet model and estimate uncertainty for classification task running on GPU |
| 3 | Mid term report | Train quantized DNN models and estimate uncertainty for classification task running on edge devices |
| 4 | Regression task | Train quantized DNN models and estimate uncertainty for regression task running on edge devices |
| 5 | Segmentation task | Train quantized ResNet model and estimate uncertainty for segmentation task running on edge device |
| 6 | Project report | Document all findings and submission of final report |

Table 1: Project milestones

## 3.3 Project Schedule



| | | Start Date | End Date |
|---|---|---|---|
| **WP 1** | **Literature search** | | |
| 1.1 | Taxonomy of NN compression methods | 15/12/22 | 15/01/23 |
| 1.2 | Quantization aware training for DL models | 15/12/22 | 15/01/23 |
| 1.3 | Uncertainty estimation evaluation methods | 15/12/22 | 15/01/23 |
| 1.4 | Estimating uncertainty for deep learning models | 15/12/22 | 15/01/23 |
| **WP2** | **Experimental setup and analysis** | | |
| 2.1 | Implement and perform experiment for ResNet model with single pass UE methods on GPU | 01/01/23 | 20/01/23 |
| 2.2 | Implement and perform experiment for ResNet model with quantization & single pass UE methods on GPU for classification task | 17/01/23 | 30/01/23 |
| **WP3** | **Mid term report** | | |
| 3.1 | Compile and deploy quantized ResNet model on edge device to estimate model uncertainty for classification task | 01/02/23 | 15/02/23 |
| 3.2 | Implement, compile and deploy quantized MobileNet model on edge device to estimate model uncertainty for classification task | 15/02/23 | 15/03/23 |
| 3.3 | Comparative evaluation of the model uncertainty estimates for classification task | 01/03/23 | 30/03/23 |
| **WP4** | **Regression task** | | |
| 4.1 | Implement, compile and deploy quantized DNN models on edge device to estimate model uncertainty for regression task | 27/03/23 | 27/04/23 |
| 4.2 | Comparative evaluation of the model uncertainty estimates for regression task | 20/04/23 | 05/05/23 |
| **WP5** | **Segmentation task** | | |
| 4.1 | Implement, compile and deploy quantized DNN models on edge device to estimate model uncertainty for segmentation task | 01/05/23 | 20/05/23 |
| 4.2 | Comparative evaluation of the model uncertainty estimates for segmentation task | 17/05/23 | 07/06/23 |
| **WP6** | **Project report** | | |
| 6.1 | Project report | 15/12/22 | 15/08/23 |

Figure 4: Project schedule

## 3.4 Deliverables

### Minimum Viable

- Taxonomy of neural network compression methods for deep learning model
- Train ResNet model with a single pass uncertainty estimation method on ImageNet dataset for classification task
- Perform experiments to estimate ResNet model uncertainty for classification task on GPU
- Train ResNet and MobileNet models with quantization techniques and single pass uncertainty estimation methods on ImageNet dataset for classification task
- Compile quantized ResNet and MobileNet model to support deploying on Google Coral - USB Accelerator and OAK-D
- Perform experiments to estimate quantized ResNet and MobileNet models uncertainty for classification task on Google Coral - USB Accelerator and OAK-D devices
- Comparative evaluation of the quantized model uncertainty estimates for classification task with difference in model, edge devices and single pass uncertainty estimation methods

### Expected

- Train ResNet and MobileNet models with quantization techniques and single pass uncertainty estimation methods on Facial Keypoints dataset for regression task
- Compile quantized ResNet and MobileNet model to support deploying on Google Coral - USB Accelerator and OAK-D
- Perform experiments to estimate quantized ResNet and MobileNet models uncertainty for regression task on Google Coral - USB Accelerator and OAK-D devices

- Comparative evaluation of the quantized model uncertainty estimates for regression task with difference in model, edge devices and single pass uncertainty estimation methods

### Desired

- Train ResNet model with quantization techniques and a single pass uncertainty estimation method on RoboCup@Work dataset for segmentation task
- Compile quantized ResNet model to support deploying on Google Coral - USB Accelerator
- Perform experiments to estimate quantized ResNet model uncertainty for segmentation task on Google Coral - USB Accelerator
- Evaluation of the quantized model uncertainty estimates for segmentation task

# References

[1] Jan Achterhold, Jan Mathias Koehler, Anke Schmeink, and Tim Genewein. Variational network quantization. In *International Conference on Learning Representations*, 2018.

[2] Andrei Atanov, Arsenii Ashukha, Dmitry Molchanov, Kirill Neklyudov, and Dmitry Vetrov. Uncertainty estimation via stochastic batch normalization. 2018.

[3] Murat Seckin Ayhan and Philipp Berens. Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks. 2018.

[4] Rainer Bischoff, Ulrich Huggenberger, and Erwin Prassler. Kuka youbot-a mobile manipulator for research and education. In *IEEE International Conference on Robotics and Automation*, 2011.

[5] Intel Corporation. neural-compressor. URL `https://github.com/intel/neural-compressor`. [Online; accessed 30-November-2022].

[6] Martin Ferianc, Partha Maji, Matthew Mattina, and Miguel Rodrigues. On the effects of quantisation on model uncertainty in Bayesian neural networks. In *Conference on Uncertainty in Artificial Intelligence*, 2021.

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

[8] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. 2021.

[9] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in Neural Information Processing Systems*, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. 2017.

[12] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 2017.

[13] kaggle. Facial Keypoints Dataset. URL `https://www.kaggle.com/datasets/neelaryan/facial-keypoints-dataset`. [Online; accessed 30-November-2022].

[14] Mina Khoshdeli, Richard Cong, and Bahram Parvin. Detection of nuclei in HE stained sections using convolutional neural networks. In *IEEE International Conference on Biomedical & Health Informatics*, 2017.

[15] Gerhard K Kraetzschmar, Nico Hochgeschwender, Walter Nowak, Frederik Hegger, Sven Schneider, Rhama Dwiputra, Jakob Berghofer, and Rainer Bischoff. Robocup@work: competing for the factory of the future. Springer, 2014.

[16] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 2017.

[17] Leandro Ariel Libutti, Francisco D Igual, Luis Pinuel, Laura De Giusti, and Marcelo Naiouf. Benchmarking performance and power of USB accelerators for inference with MLPerf. In *Accelerated Machine Learning*, 2020.

[18] MAS-Group. Robocup@work, 2019. URL `https://mas-group.inf.h-brs.de/?page_id=23`.

[19] Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning. International Joint Conferences on Artificial Intelligence, 2017.

[20] Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty Quantification in Scientific Machine Learning: Methods, Metrics, and Comparisons, 2022.

[21] Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, Ramesh Chukka, Cody Coleman, Sam Davis, Pan Deng, Greg Diamos, Jared Duke, Dave Fick, J. Scott Gardner, Itay Hubara, Sachin Idgunji, Thomas B. Jablin, Jeff Jiao, Tom St. John, Pankaj Kanwar, David Lee, Jeffery Liao, Anton Lokhmotov, Francisco Massa, Peng Meng, Paulius Micikevicius, Colin Osborne, Gennady Pekhimenko, Arun Tejusve Raghunath Rajan, Dilip Sequeira, Ashish Sirasao, Fei Sun, Hanlin Tang, Michael Thomson, Frank Wei, Ephrem Wu, Lingjie Xu, Koichi Yamada, Bing Yu, George Yuan, Aaron Zhong, Peizhao Zhang, and Yuchen Zhou. MLPerf Inference Benchmark, 2019.

[22] Casey Ross and Ike Swetlitz. Ibms watson supercomputer recommended unsafe and incorrect cancer treatments, internal documents show. *Stat News*. URL `https://www.statnews.com/wp-content/uploads/2018/09/IBMs-Watson-recommended-unsafe-and-incorrect-cancer-treatments-STAT.pdf`. [Online; accessed 30-November-2022].

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[24] Murat Sensoy, Melih Kandemir, and Lance M. Kaplan. Evidential Deep Learning to Quantify Classification Uncertainty. *CoRR*, 2018.

[25] Anna Syberfeldt and Fredrik Vuoluterä. Image processing based on deep neural networks for detecting quality problems in paper bag production. *College International pourla Recherche en Productique*, 2020.

[26] Joost van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Simple and Scalable Epistemic Uncertainty Estimation Using a Single Deep Deterministic Neural Network. *CoRR*, 2020.