2022 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT)
March 10-11, 2022. Texas A&M University - EnMed, Houston, TX, USA

2022 IEEE Healthcare Innovations and Point of Care Technologies (HI-POCT) | 978-1-6654-9615-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/HI-POCT54491.2022.9744071

Leapfrogging Medical AI in Low-Resource Contexts Using Edge Tensor Processing Unit

Priyanshu Sinha[1] and Judy W. Gichoya[2] and Saptarshi Purkayastha[1]

***Abstract*— With each passing year, the state-of-the-art deep learning neural networks grow larger in size, requiring larger computing and power resources. The high compute resources required by these large networks are alienating the majority of the world population that lives in low-resource settings and lacks the infrastructure to benefit from these advancements in medical AI. Current state-of-the-art medical AI, even with cloud resources, is a bit difficult to deploy in remote areas where we don't have good internet connectivity. We demonstrate a cost-effective approach to deploying medical AI that could be used in limited resource settings using Edge Tensor Processing Unit (TPU). We trained and optimized a classification model on the Chest X-ray 14 dataset and a segmentation model on the Nerve ultrasound dataset using INT8 Quantization Aware Training. Thereafter, we compiled the optimized models for Edge TPU execution. We find that the inference performance on edge TPUs is 10x faster compared to other embedded devices. The optimized model is 3x and 12x smaller for the classification and segmentation respectively, compared to the full precision model. In summary, we show the potential of Edge TPUs for two medical AI tasks with faster inference times, which could potentially be used in low-resource settings for medical AI-based diagnostics. We finally discuss some potential challenges and limitations of our approach for real-world deployments.**

***Index Terms*— Model Optimization, INT8 Quantization, Edge TPU, X-Ray, Ultrasound**

I. Introduction

Machine learning infrastructure consists of an ecosystem of high powered server architectures (cloud, local or hybrid) and edge devices such as mobile phones, embedded devices and wearables. Each infrastructure has its own advantages and disadvantages. Edge devices are usually constrained in terms of memory and computing power, but provide the benefit of being cheap, and not limited by network latency. This makes edge devices accessible in limited resource settings or where there is limited ability to transfer medical information due to strict regulations around medical data use. With improved performance of state of the art (SOTA) medical algorithms over time, there is a concurrent increase in the computation demand of these models [1]. This places a significant resource-burden to implement these models, with large corporations developing chips for their own internal use. Edge devices can help overcome this financial limitation, for example high end GPU servers cost more than 1000 USD, whereas the Google Coral board [2] (the edge TPU we used) costs around 130 USD.

[1] School of Informatics and Computing, Indiana University-Purdue University, Indianapolis, IN 46202, USA. {prisinha, sapturk}@iu.edu

[2] Emory University School of Medicine, Emory University, Atlanta, GA 30322, USA. judywawira@emory.edu

Model optimization is the compression of deep learning models so that they can fit into resource-constrained devices and consume less energy. There exists three approaches to optimization for edge devices: (a.) connection pruning [3], (b.) INT8 quantization [4], and (c.) knowledge distillation [5]. For our experiment, we used INT-8 quantization which is an optimization technique that converts 32-bit floating point numbers (weights and activations) to 8-bit integers. This produces a model with smaller memory footprint and reduced latency on low-power devices such as microcontrollers and integer only accelerators such as Coral edge TPU. As shown in this paper and related works, this reduction in precision often has little impact on model accuracy, but reduces memory usage up to 4x (such as when reducing 32-bit floats to 8-bit integers). INT-8 quantization can be performed in two ways, post training quantization or quantization aware training. Quantization is a lossy process that can impact the performance of a model. This loss can be minimized using Quantization Aware Training (QAT) [6].

In this paper, we compiled our models for edge TPU and evaluated their performance. We present our methodology and results in the next sections, followed by a discussion of

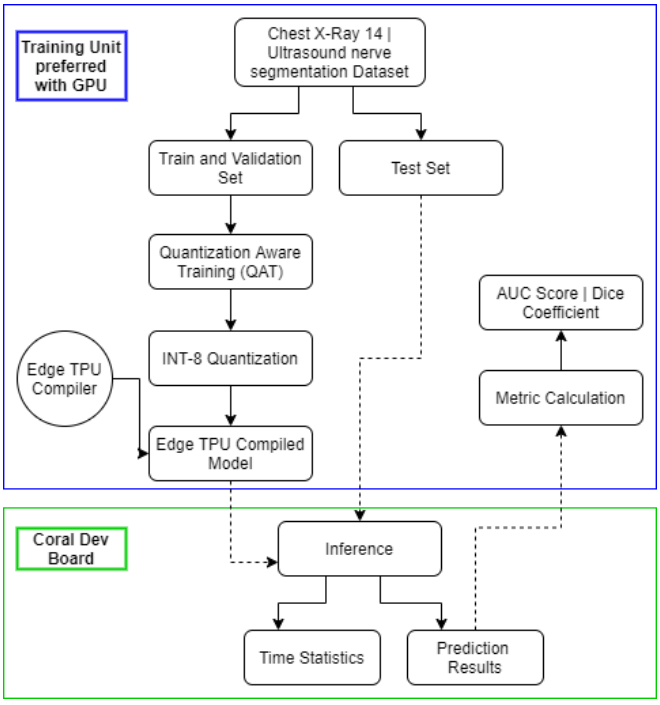

Fig. 1. Process of Evaluating Edge TPU Model

our work in relation to current literature and future scope.

II. Methodology

To evaluate the performance of edge TPU, we performed classification and segmentation task on multiple modalities of medical images i.e., x-rays and ultrasound images. Our selection of these common tasks, on widely available modality, datasets and common neural network architectures was done to identify the generalizability of our approach.

A. Classification

For the classification task, we used the NIH-14 Chest-Xray 14 dataset [7] which consists of 112,120 x-ray images of 30,805 unique patients. Each image has 14 pathology labels. We split the dataset into train (54,091 images), validation (23,183 images) and test (33,118 images) and downscaled images to 224x224. The baseline model is 32-bit floating point chest x-ray classification model, which is based on Densenet-121 architecture. This model is trained by Arevalo and Beltran [8]. Densenet is a fairly common architecture for classification task in medical imaging, which is something we wanted to use to verify our quantization. There maybe other complex architectures that may not be suitable for QAT, particularly INT8 quantization.

B. Segmentation

For segmentation task, we used the Ultrasound nerve segmentation dataset from 2016 Kaggle challenge [9]. This dataset contains 11,143 images, which is further split into 5635 training images and 5508 testing images. The images were preprocessed and downscaled to 80x112 for training and evaluation. The baseline model for this task is 32-bit floating point is based on the UNet architecture [10]. We used the model trained by George Batchkala, where he used a customized approach for UNet model [11]. UNet is a popular segmentation architecture in medical imaging and we wanted to validate INT8 inference performance of this widely used architecture.

C. Model Optimization

Since our goal is to run the model on edge TPU, a resource-constrained device, we optimized FP-32 models to INT8 using QAT. INT8 quantization is an optimization technique used to convert 32-bit floating-point numbers (weights and activation) to 8-bit integers. This reduces the model's size and reduces its latency on low-powered devices such as microcontrollers and integer-only accelerators like Edge TPU.

Since the direct reduction of precision from 32-bit floating to 8-bit integers is a lossy process, this can impact the model's performance. To minimize this loss, we used Quantization Aware Training (QAT). This simulates low precision behavior in forward pass, whereas the backward pass remains the 32-bit precision floating point. This causes some quantization error accumulated in total loss. The optimizer tries to reduce this error by adjusting its parameters and thus making it more robust to quantization, and almost lossless [12].

For the optimization, we used Tensorflow model optimization toolkit [13]. We performed the QAT for classification as well as segmentation tasks. The Densenet-121 [14] model is finetuned from weights for Imagenet [15]. We used Adam optimizer [16] with an initial learning rate of 0.001 and batch size of 32 with image generators. The UNet model [10] is trained from scratch with an initial learning rate of 0.00001 and Adam optimizer. For the UNet model training, we selected a batch size of 128 with early stopping.

The quantization aware training is performed on a GPU server. This model is further quantized to 8-bit integer. The edge TPU compiler [17] is then used to compile both the INT8 models to an edge TPU compatible model. When compiling, we didn't use the *num_segments* parameter, as we are only using one TPU and not pipelining our model for multiple TPUs.

D. Evaluation

After compiling the model for edge TPU, we evaluated its performance on Coral Dev board, and Nvidia Jetson Nano [18]. Since TensorFlow is not natively supported on the Coral board, we copied the test set to the Coral board and evaluated the model. We computed the time taken on the Coral board and serialized the prediction results using JSON file where image path is the key and inference result is the value corresponding to that for Chest X-rays and CSV file for Nerve segmentation. We copied the inference result back to the server and computed the AUC score and AUC-ROC curve for chest x-rays and the Dice score for nerve segmentation.

We used the test set of each dataset, i.e., 33,118 images of chest x-rays and 5508 images for nerve segmentation, and computed the overall time and average time for inference. We only included time for model prediction and did not include other timings such as image processing and interpreter invocation. Figure 1 explains the process we followed for evaluating the timing and AUC score for the edge TPU compiled model.

III. Results

The inference result on Coral dev board for classification and segmentation seems promising. The tables below present the various perspective of edge TPU usage in terms of model performance, inference latency and model size.

A. Performance

For segmentation, the Dice coefficient for FP32 baseline model is 0.617, whereas for the edge TPU model its 0.646. Table I summarises this performance.

Metric	Baseline FP 32	Edge TPU Model
Dice Coefficient	0.617	0.646

TABLE I
PERFORMANCE OF FP-32 AND EDGE TPU MODEL ON SEGMENTATION TASK

Class	Baseline FP 32	Edge TPU Model
Atelectasis	0.78	0.75
Cardiomegaly	0.90	0.85
Consolidation	0.79	0.77
Edema	0.88	0.85
Effusion	0.87	0.86
Emphysema	0.88	0.83
Fibrosis	0.79	0.77
Hernia	0.83	0.77
Infiltration	0.71	0.69
Mass	0.82	0.78
Nodule	0.73	0.66
Pleural Thickening	0.77	0.73
Pneumonia	0.74	0.70
Pneumothorax	0.85	0.82
Mean AUC-ROC	**0.81**	**0.77**

TABLE II

AUC SCORES COMPARISON ON CLASSIFICATION TASK

Architecture	Baseline FP 32	Edge TPU Model
Model Size (MB)	27.9	8.4
Size Reduction	-	3x

TABLE III

CLASSIFICATION MODEL SIZE REDUCTION

The model compiled for edge TPU shows minor decrease in AUC-ROC score with comparison to the full precision model. The AUC score of edge TPU model is 0.77, whereas the FP32 model has the AUC score of 0.81. The Table II shows the comparison of FP32 and edge TPU compiled model for all the 14 classes present in the dataset.

B. Model Size

The size of our baseline FP-32 model is 27.9 MB and the size of edge TPU compiled model is 8.4 MB leading to a decrease of 3x for classification task (Table III), whereas for segmentation task, the model size reduced from 98.8 MB to 8.38 MB which is approx 12x reduction in size of model. (Table IV

C. Inference Latency

The Coral dev board showed remarkable improvement in the inference timing. With only one edge TPU, it took only 24 ms per image for inference for classification task and 5.44 ms for segmentation task. This is significantly less than the other embedded board, i.e., Nvidia Jetson Nano. Table V and Table VI summarizes this.

IV. CONCLUSION AND FUTURE SCOPE

Edge computing is changing the way data is handled and transported. It's a low-latency and cost efficient solution to

Architecture	Baseline FP 32	Edge TPU Model
Model Size(MB)	98.8	8.38
Size Reduction	-	12x

TABLE IV

SEGMENTATION MODEL SIZE REDUCTION

Devices	INT8	Edge TPU Model
Nvidia Jetson Nano	410	NA
Coral Dev Board	NA	**24**

TABLE V

INFERENCE TIME FOR CLASSIFICATION IN MILLISECONDS (MS/IMAGE)

Devices	INT8	Edge TPU Model
Nvidia Jetson Nano	309	NA
Coral Dev Board	NA	**5.44**

TABLE VI

INFERENCE TIME FOR SEGMENTATION IN MILLISECONDS (MS/IMAGE)

Medical AI inference. Machine learning at edge is gaining traction in recent days [19] as it can solve various issues such as data privacy, low latency solution, etc. Various companies such as Google (edge TPU), Nvidia (Xavier and Jetson), Intel (Neural Compute Stick) are investing into more efficient hardware, better optimizers and compilers.

Wisultschew et. al [20] performed an study and compared the efficiency and performance of Google's Edge TPU and the Intel Neural Compute Stick for 3D object detection. Reuther et. al.[21] compared the performance of the Edge TPU to an standard Intel Core i9 CPU and Intel's second version of Neural Compute Stick and it is found that edge TPU performed comparable to standard CPU while consuming less power. Kljucaric et al. [22] found that GoogleNet on Edge TPU outperforms NVIDIA Xavier and NCS2 in optical character recognition task. This [23] explored the trade-off between computational and energy efficiency feed-forward and convolutional neural networks for edge TPU and cortex A53 platforms. Google research team evaluated the performance of edge TPU on NASBench dataset with 423k different neural architectures and studied the latency and accuracy of different models. This [24] explored the performance of edge TPU in terms of sensitivity to the variations of model's architecture and specifications.

Here, we have extended our previous work of medical imaging model optimization [25] and presented the use case of edge TPU using Coral dev board. We found that there is negligible loss in accuracy of model i,e., 0.04 in mean AUC-ROC score for chest x-rays and it performed better for segmentation task (dice coefficient of 0.646 for edge TPU and 0.618 for baseline model). The inference timing of edge TPU model is 24 ms. Also, the size of edge TPU model is 3x less than the full precision model making it suitable to fit in the memory of resource constrained device. We believe that with the edge TPU we can deliver medical AI in low-resource contexts at a low cost. Medical AI has potential to support low-skilled health workers [26], or reduce the burden of fewer specialists in low-resource context, including opportunities for lowering costs in high-resource contexts through reverse innovation [27].

With the advent of edge TPU accelerators, many devices can use a pluggable device for accelerating the inference of machine learning models. Devices with on-board edge TPUs

like Asus Tinker Edge R [28], Imagio Vision Cam AI [29], Pixel phones [30] and others can collect health data from wearables, smartphone cameras and perform on-device AI inference to alert the user in case of any abnormality, as well as processing of critical PHI data on-device without leaving the hospital network. Further, in make-shift hospitals, which are common in disaster situations, we can build edge TPU server farms that can be suited to run off battery supplies and still provide access to medical AI. This can be also used for ultrasound imaging where we have handheld ultrasound device from GE healthcare [31] or Butterfly network [32].

Though edge TPU model had good performance on the Coral board, there are some critical limitations. Images needed to be pre-processed due to lack of library support on the Coral board before inference. We calculated the time for inference but additional I/O activities should also be considered in the overall time. In the future, we will measure the thermal efficiency of Coral board and evaluate other important imaging tasks such as localization, image generation, and use of other neural network architectures.

V. Source Code

The source code of the experiments can be found at: https://github.com/pri2si17-1997/Edge-TPU-Evaluation

References

[1] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-lm: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv:1909.08053*, 2019.

[2] "Coral dev board." [Online]. Available: https://coral.ai/products/dev-board

[3] N. Lee, T. Ajanthan, and P. H. S. Torr, "Snip: Single-shot network pruning based on connection sensitivity," 2019.

[4] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.

[5] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015.

[6] G. LLC, "Quantization aware training," https://www.tensorflow.org/model_optimization/guide/quantization/training.

[7] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.369

[8] WillArevalo and J. Beltran, "xrays-multi-densenet-121," https://www.kaggle.com/willarevalo/xrays-multi-densenet121, 2019.

[9] Kaggle., "Ultrasound nerve segmentation," https://www.kaggle.com/c/ultrasound-nerve-segmentation/data.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[11] G. Batchkala, 2019. [Online]. Available: https://www.kaggle.com/gbatchkala/urss-2019-project-review

[12] V. Nandwani, "Inside quantization aware training," https://towardsdatascience.com/inside-quantization-aware-training-4f91c8837ead, 2021.

[13] "Tensorflow model optimization." [Online]. Available: https://www.tensorflow.org/model_optimization

[14] 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.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] G. LLC., "Edge tpu compiler," https://coral.ai/docs/edgetpu/compiler/.

[18] N. Corporation, "Jetson nano developer kit," https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[19] S. Hosseininoorbin, S. Layeghy, B. Kusy, R. Jurdak, and M. Portmann, "Exploring deep neural networks on edge tpu," 2021.

[20] C. Wisultschew, A. Otero, J. Portilla, and E. de la Torre, "Artificial vision on edge iot devices: A practical case for 3d data classification," in *2019 XXXIV Conference on Design of Circuits and Integrated Systems (DCIS)*, 2019, pp. 1–7.

[21] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and benchmarking of machine learning accelerators," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, 2019, pp. 1–9.

[22] L. Kljucaric, A. Johnson, and A. D. George, "Architectural analysis of deep learning on edge accelerators," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, 2020, pp. 1–7.

[23] S. Hosseininoorbin, S. Layeghy, M. Sarhan, R. Jurdak, and M. Portmann, "Exploring edge tpu for network intrusion detection in iot," 2021.

[24] A. Yazdanbakhsh, K. Seshadri, B. Akin, J. Laudon, and R. Narayanaswami, "An evaluation of edge tpu accelerators for convolutional neural networks," 2021.

[25] A. Abid, P. Sinha, A. Harpale, J. Gichoya, and S. Purkayastha, "Optimizing medical image classification models for edge devices," in *International Symposium on Distributed Computing and Artificial Intelligence*. Springer, 2021, pp. 77–87.

[26] S. L. Bucher, P. Cardellichio, N. Muinga, J. K. Patterson, A. Thukral, A. K. Deorari, S. Data, R. Umoren, and S. Purkayastha, "Digital health innovations, tools, and resources to support helping babies survive programs," *Pediatrics*, vol. 146, no. Supplement 2, pp. S165–S182, 2020.

[27] S. N. Kasthurirathne, B. W. Mamlin, S. Purkayastha, and T. Cullen, "Overcoming the maternal care crisis: how can lessons learnt in global health informatics address us maternal health outcomes?" in *AMIA Annual Symposium Proceedings*, vol. 2017. American Medical Informatics Association, 2017, p. 1034.

[28] [Online]. Available: https://tinker-board.asus.com/product/tinker-edge-r.html

[29] "Smart ai camera coral camera visionai - imago-technologies," Oct 2021. [Online]. Available: https://imago-technologies.com/smart-camera-with-deep-learning-accelerator/

[30] "Pixel 6." [Online]. Available: https://store.google.com/product/pixel_6?hl=en-US#p6-overview-whitechapel

[31] G. E. Company, "Point of care ultrasound," https://www.gehealthcare.com/products/ultrasound/point-of-care-ultrasound.

[32] B. N. Inc., "One probe, whole body imaging," https://www.butterflynetwork.com/iq.