Comparing Quantization Methods for On-Edge ECG Interpretation using Multi-Task CNN

Kiriaki J. Rajotte
Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, MA, USA
kjrajotte@wpi.edu

Bashima Islam
Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, MA, USA
bislam@wpi.edu

David D. McManus
Department of Medicine
University of Massachusetts Chan Medical School
Worcester, MA, USA
david.mcmanus@umassmed.edu

Xinming Huang
Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, MA, USA
xhuang@wpi.edu

Edward A. Clancy
Electrical and Computer Engineering
Worcester Polytechnic Institute
Worcester, MA, USA
ted@wpi.edu

***Abstract*—Wearable devices have begun to incorporate machine learning models to assist with detection of various cardiac conditions. In this work, we developed a multi-task convolutional neural network to simultaneously predict 75 diagnostic, form and rhythm statements from 10-s duration, 12-lead ECGs. The model, originally developed off-line in TensorFlow, was converted to the FlatBuffers format for on-edge AI using the LiteRT toolset. Post-training quantization was used to compare different numerical precisions in terms of model size, model performance and inference time. Classifier performance for the 12-lead configuration was consistent between the 32-bit floating point model ("float32" baseline), the dynamic range quantized model (DR) and the float16 model (p=0.92) with an average macro AUC score of 0.893 with all output statements considered. A large degradation in classification performance was observed for 8-bit integer quantization (int8) which yielded an average macro AUC score of 0.513 for the 12-lead configuration across all statements. To address class imbalance, minority classes were removed. Reducing the number of statements to 41 classes increased macro F1 score by an average of 72.6% (to a mean value about 0.358) for the float32, float16 and DR quantized models.**

Keywords—ECG interpretation, Embedded Machine Learning, Multi-label Classification, Electrocardiogram, LiteRT

I. Introduction

Since 1921, the United States Centers for Disease Control and Prevention has identified heart disease as the leading cause of death in the United States. This trend continues over 100 years later with a 2024 report noting that 48.6% of individuals in the United States have some type of cardiovascular disease such as coronary heart disease, heart failure, stroke and/or high blood pressure. Over time, improvements in diagnostics and available treatments have helped to reduce morbidity and mortality [1]. In recent years, clinical and consumer grade wearable devices have also come to play a role in the detection of some cardiovascular concerns through remote monitoring of the electrocardiogram (ECG). Clinical grade wearables, such as Holter monitors and event monitors, have enabled remote patient monitoring over the span of a few hours and in some cases up to a month, potentially capturing irregularities in the electrical activity of the heart. These devices allow clinicians to record intermittent irregularities/arrhythmias that may not have been present during an in-office patient visit. With the increased available ECG data also comes greater interpretation burden on the clinicians. To alleviate this burden, machine learning and AI tools have been implemented to assist.

There are many classification models found in the literature and integrated into consumer devices; many of these models focus on the binary classification of abnormal or not, while most others focus on detecting a specific condition or small group of conditions. While these classifiers are helpful in identifying a concern and will hopefully lead to appropriate interventions, they are limited in their diagnostic contribution and often require further testing. With the advent of machine learning and AI, deeper machine learning models have become more widely available, which in turn enables more complex classifiers that may provide more information about possible conditions. However, adoption of such models is limited by their large computational and memory costs. This work investigates the impact of standard post-training quantization approaches on the performance of a multi-task convolutional neural network (MT-CNN) trained to identify ECG statements from three distinct groups of conditions from the multi-label PTB-XL dataset [2], [3], [4], for different number of inputs (ECG leads) and outputs (classes considered). The goal of this work is to show that existing post-training quantization approaches are sufficient in enabling effective, multi-task, on-device classification of ECG statements without compromising performance. The use of post-training quantization without calibration or retraining provides a valuable performance baseline that can inform future deployment or optimization efforts.

II. Related Works

Numerous algorithms have been proposed over the years to aid in the interpretation of ECG signals. Since its release, the MIT-BIH arrhythmia dataset has enabled researchers to

develop algorithms and machine learning models to identify different arrythmias. Other datasets, both private and public, such as the PTB and PTB-XL, cover conditions beyond arrhythmias [2], [5]. In conjunction with the growth of wearable and IoT devices, these models are becoming more computationally efficient, enabling on-edge computation which comes with benefits such as improved security, reduced reliance on wireless networking and longer-term monitoring for clinical and consumer applications.

Xie and Lin proposed a novel model called YOLO-ResTinyECG for arrhythmia detection using ECG images [6]. Their model incorporates YOLO object detection with a lightweight residual network and utilizes a weighted cross-entropy loss function during training to improve classification of minority arrhythmia classes. Their models were trained to detect 6 or 9 possible arrhythmias with varying window lengths and then deployed to a Jetson Nano Development Kit (NVIDIA, Santa Clara, USA). They achieved a mean average precision (mAP) of 92.35% (F1 score of 0.89) for a 6 class detector and a mAP of 90.2% (F1 score of 0.82) for a 9 class detector. The aims of their work align with the work described herein, but focus on arrhythmia-only detection, while our work also includes form and diagnostic classification.

Kim et al. [7] utilized the MIT-BIH arrhythmia and PTB diagnostic datasets to train a CNN model to detect normal or "abnormal" ECG in a real-time embedded system using the Arduino Nano BLE Sense development kit. Their model consists of two CNN blocks (a convolutional layer, max. pooling and dropout) followed by a flattening layer and then two dense layers. To convert their model to run on a microcontroller, the authors used TensorFlow Lite for Microcontrollers, the toolset preceding LiteRT (LiteRT was used in this work). Their 27 kB lite model achieved an accuracy of 97% when tested on the PTB test data. Their work demonstrates the ability to run a lightweight CNN model in an embedded environment, but is limited in the information output from the model as no specific details are provided.

In this work, a MT-CNN was trained and then quantized using LiteRT to evaluate its ability to classify all 75 statements in the multi-label PTB-XL dataset for a resource constrained environment such as wearables or remote patient monitoring devices. While many prior works have focused on the classification of fewer classes, such as the classification of only arrhythmias [6] or normal vs. "abnormal" [7], this model is able to classify diagnostic, form and rhythm statements. In a wearable ECG monitoring system, minimizing the number of physical electrode connections to the subject can make the system easier to wear, so this model was also trained for 3 different combinations of ECG leads to study the influence of using a reduced ECG lead set on model size, inference time and classification performance. Additionally, to address class imbalance, minority classes were excluded, and performance was re-considered. Evaluation of lead- and class-subsets are practical steps towards on-device deployment to an embedded wearable device in which a reduced lead set is more accessible. LiteRT was used to quantize and evaluate the model since it enables the deployment of TensorFlow models to a large set of possible target environments such as Raspberry Pi, microcontrollers, or iOS and Android [8].

III. Methods

A. Data Used

The publicly-available PTB-XL dataset consists of 21,837 12-lead ECGs measured from 18,885 patients [2], [3], [4]. The PTB-XL recordings were sampled at 500 Hz over a 10 second duration. This ECG dataset was annotated by up to two cardiologists and contains ECG statements spanning 44 diagnostic (e.g., normal, hypertrophy, myocardial infarction), 19 form (e.g., abnormal QRS, digitalis-effect) and 12 rhythm categories (e.g., atrial fibrillation, sinus tachycardia, atrial flutter). Four of the form statements overlap with the diagnostic statements. In this work, the train (80%), validation (10%) and test split (10%) recommended by the authors of [3] was used as they provided splits that balanced label, age and sex distribution. All leads were standardized to zero mean and unit variance.

B. Classification Model Architecture

A MT-CNN was used to classify the statements in the PTB-XL dataset. The MT-CNN consists of six shared convolutional blocks (with filter sizes of {128, 128, 256, 256, 128, 128}) built from a convolutional layer, batch normalization layer and rectified linear unit (ReLU) activation function. Following the six convolutional blocks is a global average pooling layer. After this pooling layer, the model splits into the tasks of classifying the 44 diagnostic, 19 form and 12 rhythm statements. Each output path consists of two dense layers, the first utilizes a ReLU activation and the second utilizes a sigmoid activation function to generate the likelihood of each statement. The number of parameters varies based on the number of inputs and outputs, ranging from 1,709,741 for the model with 3 inputs and 41 outputs up to 1,728,971 for the model with all inputs and outputs. When training the model, the ADAM optimizer was used with a learning rate of 0.001 to train over 30 epochs with a batch size of 128. The focal loss function was used to address class imbalance at the algorithmic level [9]. Training was conducted using an NVIDIA A100-SXM4-80GB graphics processing unit.

C. Varying Number of Input Leads and Number of Outputs

A clinical ECG recording typically uses a standardized set of 12-leads measured using electrodes placed across the upper body to ensure sufficient spatial measurement resolution [10]. In a clinical setting, capturing a 12-lead ECG usually takes only a few minutes. While 12-lead ECGs can provide a quick, non-invasive look at the electrical activity of the heart, they can be cumbersome to capture outside of the clinical setting. For clinical grade remote patient monitors such as the Holter monitor or event monitors, and consumer wearables, it is common to acquire only a subset of ECG leads or a single lead. In such devices, computational resources may be constrained and, practically, it is inconvenient for users to have multiple wires connected to their chest while trying to perform day-to- day activities. In this study, the classification model was trained for three different sets of ECG inputs: a full 12-lead; a 5-lead subset consisting of Leads I, II, V1, V3, and V6; and a 3-lead subset consisting of Leads I, II, and V2. Selection of these lead subsets was made based on

preliminary model analysis in which it was observed that the classification performance was maintained as the number of inputs was reduced from 12-leads down to as few as 3-leads [11].

To address class imbalance in the PTB-XL dataset, the classes with the fewest samples were then excluded when training the model. Removal of underrepresented classes was pursued as it is a conservative approach to address class imbalance in the absence of additional training data. The use of data augmentation, data resampling or more robust training schemes could alleviate the effects of class imbalance but at the risk of introducing bias into the model. Model performance was compared with all outputs, 56 outputs (smallest 8 diagnostic, 4 form and 3 rhythm classes removed) and 41 outputs (smallest 16 diagnostic, 8 form and 6 rhythm classes removed). A minimum of 50% of the original outputs were retained.

D. Model Quantization and Evaluation

Post-training quantization was employed to compare the model performance achieved at various levels of precision. When using post-training quantization, the model was trained without any awareness of the intent to quantize and the saved model was quantized during conversion to the FlatBuffer model format for on-edge computing [8], [12]. Three quantization schemes were tested: dynamic range (DR), float16 and integer (int8) quantization. DR quantization converts weights to 8-bit integers statically at conversion time. The activations are stored as floating point values but are dynamically quantized to 8-bit precision during inference if required. Float16 quantization quantizes the original 32-bit floating point weights to 16-bit floating point weights, effectively cutting the model size in half. Int8 quantization converts 32-bit floating point weights and activations to the nearest 8-bit fixed point numbers. The unquantized LiteRT model was our measure of baseline performance. This model maintains the default TensorFlow precision of 32-bit floats but in the LiteRT FlatBuffer format [8].

The quantized models were evaluated in Python using the LiteRT Python Interpreter. To measure the classifier's test set performance after quantization, the macro area under the receiver operating characteristic curve (AUC) was computed for each task. The change in macro F1 score was used to assess the impact of reducing the number of possible outputs as macro AUC scores are less sensitive to class imbalance [13]. To evaluate the influence of quantization, the model size in bytes and average inference time were also considered. Inference time was measured as the time required to run a single model invocation.

Statistical analyses were performed for each input-output combination considered for the class-wise AUC scores and inference times achieved during each test run. The Kolmogorov-Smirnov test indicated that the class-wise AUC scores and inference times were not normally distributed for any input-output configurations ($p < 0.001$), so the Friedman test compared model class-wise AUC scores and inference times (separately) for the four quantization schemes considered. All instances of the Friedman test indicated statistically significant differences between the four quantization schemes considered when testing

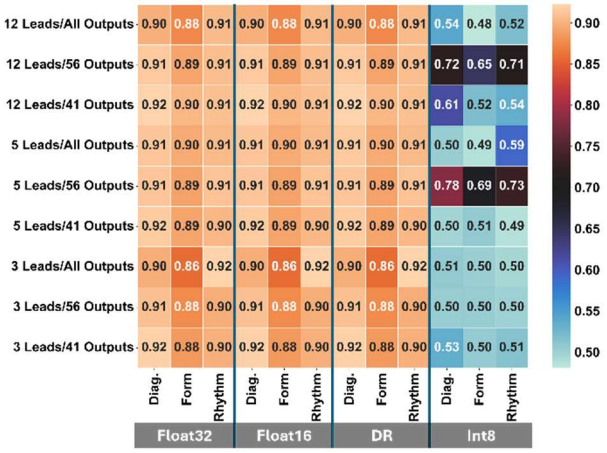

Fig. 1. Heatmap of the Macro AUC Scores for each configuration considered

the class-wise AUC scores and inference times ($p < 0.001$). Thus, *post hoc* pair-wise comparisons were performed using the Wilcoxon signed-rank test with Bonferroni-Holm correction. Results were statistically significant for a p-value less than 0.05.

IV. Results

A. Classification Performance

As shown in Fig. 1, the macro AUC scores for each output of the MT-CNN showed little to no change between the 32-bit model, the float16 quantized model and DR quantized model. No statistically significant differences were found in the pairwise comparisons of these 3 quantization schemes ($p > 0.185$) with the exception of the float32 and DR models for the {5-lead, 41 output} model ($p = 0.045$). When quantized to int8 precision, the macro AUC score dropped to ~0.5 indicating poor classification performance. Statistically significant differences were observed for all pair-wise comparisons of the int8 models to the other 3 quantizations ($p < 0.001$).

Macro AUC scores are not as sensitive to class imbalance as macro F1 scores. To assess the impact of removing the classes with the fewest samples, the change in macro F1 score was considered when removing ~20% of classes as well as when removing ~40% of classes. Results from the int8 quantization were excluded due to the model's poor performance at this precision. For the 12-lead models (Fig. 2), the macro F1 score increased by an average of 37.88% across the three output categories (from 0.149 to 0.198 for diagnostic, 0.129 to 0.182 for form, and 0.341 to 0.474 for rhythm) for the 32-bit, float16 and DR quantization when the number of outputs was reduced from 75 to 56. When the number of outputs was further reduced from all outputs down to 41, the macro F1 score increased by 72.6% (from 0.149 to 0.262 for diagnostic, 0.129 to 0.215 for form and 0.341 to 0.594 for rhythm).

B. Model Size and Inference Time

Model size decreased with decreasing model precision. The average model size (±standard deviation) for the 32-bit models is 6.549±0.028 Mb. The average model size of the float16 quantized models is roughly half that of the 32-bit model at 3.283±0.014 Mb. For the DR and int8 quantized models, the average model size is roughly a quarter of the 32-bit model at

TABLE I. INFERENCE TIME, IN MILLISECONDS (MEAN±STANDARD DEVIATION OF 2198 TEST RUNS)

Inputs/ Outputs	Quantization			
	None	*Float16*	*DR*	*Int8*
12/All	150±4	148±2	135±1	149±1
12/56	147±2	147±2	134±2	148±2
12/41	148±2	148±2	134±2	148±2
5/All	148±4	147±1	134±1	148±1
5/56	198±3	201±3	35±1	44±1
5/41	198±3	202±4	35±1	44±1
3/All	203±5	213±3.8	36±1	43±1
3/56	197±2	202±2	35±2	43±2
3/41	201±5	204±4	37±1	44±1

1.6668±0.007 Mb and 1.681±0.007 Mb, respectively.

Table I contains the average (±standard deviation) inference times for each model configuration. Statistically significant differences were detected between the inference times for all pair-wise comparisons of quantizations for each input and output scheme ($p < 0.001$). On average, the smaller DR and int8 models ran faster than the 32-bit float and float16 models.

V. DISCUSSION & CONCLUSIONS

To determine the feasibility of adapting the MT-CNN for a resource constrained application, such as a wearable ECG monitor, the model was quantized to study the impact on classification performance, model size and inference time. Classifier performance was maintained when numerical representation was compressed using the float16 and dynamic range post-training quantization, and the model's memory footprint was noticeably reduced. For applications in which memory is constrained, the DR model would likely be the most suitable as it provided the same level of classifier performance with the smallest model size. In comparison to the model in [7] which uses 27 kB/output, our DR model uses 22 kB/output. In terms of inference time, on average, smaller models had shorter inference times, but this trend may have been influenced by simultaneous processes running on the testing machine.

To improve overall model performance, a more robust data pre-processing stage should be considered. The inclusion of this stage can work to reduce noise from sources such as powerline interference, baseline wander and motion artifact, which can be detrimental in a wearable system [14]. Additionally, performance improvement may also come with techniques such as SMOTE or the augmentation of new and diverse training data to increase the number of samples for underrepresented classes and minimize class imbalance. In terms of quantization, there are two additional approaches that could be used to improve performance, particularly for int8 quantization, quantization aware training (QAT) and mixed quantization. QAT can be employed during model training to mirror inference time quantization which introduces some quantization error while the model is learning. QAT typically results in higher accuracy models when deployed, at the expense of a more complex training process. With mixed quantization, only certain layers may be quantized or distinct layers quantized to different precision depending on what is suitable for that operation. This technique would require further experimentation.

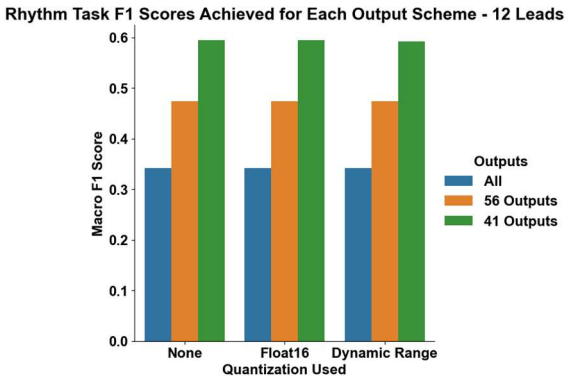

Fig. 2. Comparison of macro F1 scores for the rhythm task

In this work, the influence of quantization was explored on a MT-CNN ECG classification model to determine whether the model could be deployed to a wearable or embedded application while maintaining the ability to classify the PTB-XL's 44 diagnostic, 19 form and 12 rhythm statements. It was observed that the macro AUC score remained relatively stable (0.893) across the 32-bit, float16 quantized and DR quantized models. When using int8 quantization, the macro AUC dropped to about 0.5. In terms of model size, float16 quantization cut the model size in half and dynamic range quantization cut the model size to a quarter in comparison to the 32-bit model. With respect to inference time, the average inference time decreased with model size. Future work can explore the impact of different test environments or look to improve performance of the quantized model with the use of QAT or mixed quantization.

REFERENCES

[1] American Heart Association, "More than half of U.S. adults don't know heart disease is leading cause of death, despite 100-year reign," News Release, Jan. 2024.

[2] P. Wagner, N. Strodthoff, R.-D. Bousseljot, W. Samek, and T. Schaeffter, "PTB-XL, a large publicly available electrocardiography dataset (version 1.0.3)." PhysioNet, Nov. 09, 2022. [Online]. Available: https://physionet.org/content/ptb-xl/1.0.3/

[3] P. Wagner *et al.*, "PTB-XL: A Large publicly available ECG dataset," *Sci Data*, vol. 7, 2020.

[4] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[5] R.-D. Bousseljot, "PTB Diagnostic ECG Database." PhysioNet, Sep. 25, 2004. [Online]. Available: https://physionet.org/content/ptbdb/1.0.0/

[6] Y.-L. Xie and C.-W. Lin, "YOLO-ResTinyECG: ECG-based lightweight embedded AI arrhythmia small object detector with pruning methods," *Expert Systems with Applications*, vol. 263, Nov. 2024.

[7] E. Kim, J. Kim, J. Park, H. Ko, and Y. Kyung, "TinyML-Based Classification in an ECG Monitoring Embedded System," *Computers, Materials & Continua*, vol. 75, no. 1, pp. 1751–1764, Feb. 2023.

[8] *LiteRT overview*. (Mar. 04, 2025). Google. Accessed: Mar. 15, 2025. [Online]. Available: https://ai.google.dev/edge/litert

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, 2020.

[10] D. G. Strauss and D. D. Schocken, *Marriott's Practical Electrocardiography*, 13th ed. Wolters Kluwer Health, 2021.

[11] K. Rajotte, "Design and Development of Intelligent, Low-Power, Wireless Wearable Sensors for Biopotential Measurement," Worcester Polytechnic Institute, 2025.

[12] TensorFlow, *Quantization aware training comprehensive guide*. (Mar. 09, 2025). Google.

[13] J. Brownlee, "Model Evaluation," in *Imbalanced Classification with Python Choose Better Metrics, Balance Skewed Classes, and Apply Cost-Sensitive Learning*, 1.2., Machine Learning Mastery, 2020.

[14] L. Sornmo and P. Laguna, "ECG Signal Processing," in *Bioelectrical signal processing in cardiac and neurological applications*, 1st ed., Elsevier Academic Press, 2005.