Phonocardiogram Segmentation with Tiny Computing

Krzysztof K. Kwiatkowski
Department of Electrical and Electronic Engineering, Imperial College London & STMicroelectronics
kkk321@ic.ac.uk

Danilo P. Pau, FIEEE
System Research and Applications
STMicroelectronics, Italy
danilo.pau@st.com

Terence Leung
Department of Medical Physics and Biomedical Engineering, University College London
t.leung@ucl.ac.uk

Oriana Di Marco
Healthcare &Wellbeing
STMicroelectronics, Italy
oriana.dimarco@st.com

***Abstract*—The stethoscope is a daily used tool that allows medical doctors to diagnose common cardiovascular diseases by listening to heart sounds. However, dedicated medical training is required to operate it. Numerous machine learning techniques have been used in attempts to automate this process and have yielded highly accurate results. However, creating a low power, portable, economical, and accurate machine learning stethoscope calls for tiny processing of phonocardiograms i.e., heart sound digital processing to run within an embedded device. To address the need to deploy the solution within a constrained tiny device, we propose an 8-bit deep learning model with low embedded FLASH and RAM utilization of 126 KiB and 45 KiB respectively, which is optimized for inference on an off-the-shelf STM32H7 microcontroller with an inference time of 12 ms, in 126KiB FLASH and 45 KiB RAM being 91.65% accurate.**

Keywords—PCG segmentation, heart sound, tiny machine learning, STM32

I. Introduction

Tiny Machine Learning (TinyML) technologies have reshaped the approaches that developers took to leverage the advantages of machine learning in consumer electronic devices. The ability to inference a deep learning model directly on small-footprint microcontrollers (MCU) removed the need for complex IoT architectures with ML workloads executed on the cloud. This helped to reduce data transfer, between the cloud and the edge device, security risks and increases scalability, all of which are key requirements to be fulfilled for consumer-grade medical devices. For users it eliminated the need for complex setup processes and the risk of connection issues while allowing more scalability and flexibility as to the conditions in which a device can be used.

Cardiovascular diseases (CVDs) are a leading cause of death in developed countries representing 32% of all global deaths [1]. Early diagnosis is key to adopt effective treatments. The popularity of consumer-grade blood pressure and ECG monitors is strong evidence that consumers are aware of the importance of monitoring blood pressure and cardiac activity on their own and daily. Therefore, they would likely also be interested in devices capable of recording and analysing phonocardiograms (PCGs) i.e., automated stethoscopes. These would have advantages over typical stethoscopes such as the ability to monitor patients for extended periods of time or during physical activity and therefore detect abnormalities which may not be evident during routine check-ups performed by doctors, who are limited in time.

PCG segmentation techniques studied in previous works are diverse and include algorithmic techniques, statistical models and deep learning models. The latter utilised convolutional neural networks (CNNs) [2], long short-term memory networks (LSTMs) [3] and temporal convolutional networks (TCNs) [4].

II. Objectives and Requirements

The goal of this study is to develop a tiny neural network model for PCG segmentation capable of inference on an ARM Cortex-M MCU and a supporting pipeline for data pre-processing and post-processing. Such a MCU can be used to produce a cheap automated stethoscope with low power consumption. The model shall achieve adequate accuracy compared to more complex pipelines or even higher to ensure that PCG segments are precisely measured. The total resource utilisation of the neural network and supporting data acquisition and processing must not exceed the FLASH and RAM memory embedded into MCU. Inference time should be minimised. Models shall prioritise high temporal resolution over PCG segmentation. It is expected that murmur analysis will take place within the same device.

III. Proposed Pipeline

Explainability is key for machine learning models adopted in medical applications. Deep learning models which aim to directly infer a diagnosis do not fit this requirement due to their black box nature. Therefore, it is required for the model to only output information that are comprehensible and verifiable by medical professional to support their process of establishing the diagnosis.

To facilitate the diagnostic procedure an automated stethoscope shall employ the pipeline shown in Figure 1.

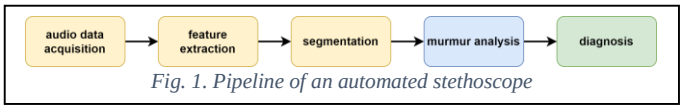

Fig. 1. Pipeline of an automated stethoscope

This paper focused on the segmentation including data pre-processing and inference stages and aimed to optimize these to fit into the smallest memory and inference time as possible without compromising the accuracy.

IV. Dataset

PCGs constitute of four distinct sections, shown in Figure 2, which occur in sequence as follows: 1st heart sound (S1), systolic interval, 2nd heart sound (S2) and diastolic interval. The time-

frequency characteristics of each of these segments and those of murmurs found in the systole and diastole, are important for the diagnosis of certain CVDs. The four sections can be derived based on the output of a machine learning segmentation model. Since the sequence is known the model shall be capable of distinguishing between the classes: S1, S2, systolic and diastolic intervals before their time-frequency characteristics can be further analysed for diagnostic purposes.

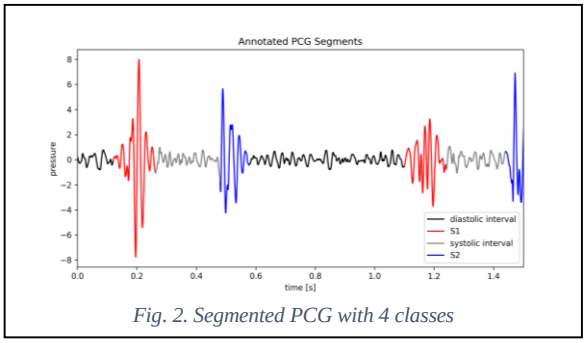

Fig. 2. Segmented PCG with 4 classes

There are many datasets available for PCG segmentation. Among them, the Logistic Regression-HSMM-based Heart Sound Segmentation dataset by David Springer available on PhysioNet [6][7] was selected by this work, although it does not include the segmented labels. It consists of 792 recordings with a sampling rate of 1 kHz from 135 patients, some of whom exhibit abnormalities in their PCGs.

V. Audio Data Pre-processing

A wide range of feature extraction techniques for PCG segmentation have been explored in related works. A Fourier synchro-squeezed transform (FSST) is an accurate technique that has been used in conjunction with a Bidirectional Long Short-Term Memory (BiLSTM) neural network [4]. This work adopted a similar approach utilising a series of short-time Fourier transforms (STFTs) for data pre-processing due to their simplicity and the availability of open-source implementations of STFT optimized for STM32 MCUs.

STFT was computed for audio data windows with a length of 128 data points sampled at 1 kHz and a stride of 1 sample. Window length was decided upon based on the clarity of the appearance of S1 and S2 in a plot of STFT magnitudes. Stride should be minimal to achieve the best possible temporal resolution. This does, however, require increased inference runs per second over the duration of PCG recording. Kaiser windowing was used and different β factors were considered (see Figures 3 and 4) [8]. It was determined that rectangular windows with β = 0 allowed the tested models to achieve higher accuracy than tapered windows with β = 5 or β = 6. It is suggested that this is due to the appearance of artefacts during S1 and S2. These created clearer boundaries between classes which reduced the misclassification rate in areas of transition.

Table 1. Comparison of accuracy between models trained using audio data pre-processed with STFT using Kaiser windowing with different values of β

Model Type	Accuracy for β = 0	Accuracy for β = 5	Accuracy for β = 6
3-class BiLSTM	**97.63%**	95.88%	95.93%
3-class fp32 CNN	**91.97%**	90.38%	90.23%

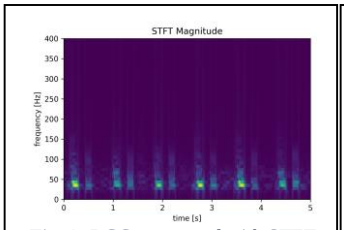

Fig. 3. PCG processed with STFT using Kaiser windowing with β=0

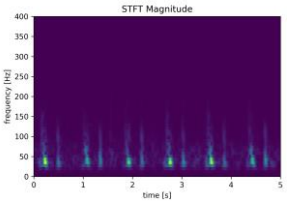

Fig. 4. PCG processed with STFT using Kaiser windowing with β=6

[8] Schmidt et. al. have determined that the frequency range of PCG features is 25 – 400 Hz. It is known that the STFT features also exist within the 20 – 25 Hz range and are faint above 200 Hz [3]. Therefore, 22 different STFT magnitudes for the 20 – 200 Hz range were inputted into the models. For BiLSTM models both the real and imaginary parts of the STFT values were included in the input array resulting in 44 values per each original audio data point. For CNN models the absolute values of the complex values were calculated resulting in 22 values per audio data point. These were inputted as an array resembling a monochromatic image. Furthermore, input values were Z-score standardised to fit the training dataset average and to avoid introducing biases into the learning.

VI. Model Topology and Hyperparameters

This work developed a model capable of inference on a tiny MCU such as, for example, the STM32H7 ARM Cortex-M7 processor, with embedded 1 MB RAM and 2 MB FLASH. The BiLSTM approach by Gaona and Arini [3] was considered as baseline and modified in early experiments. Unfortunately, it was not possible to simplify the model to fit into the MCU while maintaining sufficient accuracy. Accuracy requirement was set at 90% or above to precisely measure the duration of each S1 and S2 segment. Therefore, an alternative tinier CNN model was devised, which used the STFT based data pre-processing technique and fit onto the MCU.

Known PCG segmentation models aim to separate the signal into 4 classes: S1, systolic interval, S2, diastolic interval. It is, however, not necessary to identify systole and diastole since the sequence in which they occur is fixed. Intervals can therefore be labelled as either systolic or diastolic during data post-processing based on the occurrence of S1 and S2. Such a strategy worked well in all cases even when murmurs exist in systole and/or diastole. Therefore, a labelled PCG dataset with 3 classes was derived and an example is shown in Figure 5.

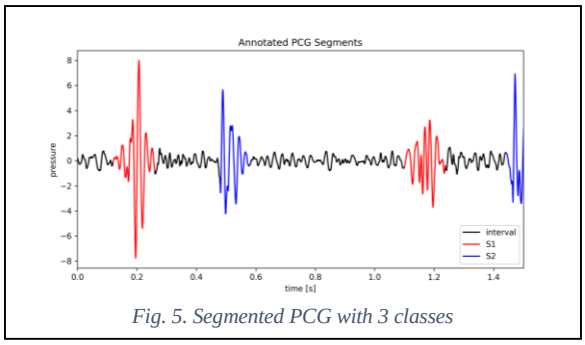

Fig. 5. Segmented PCG with 3 classes

Considering systole and diastole as the same class did not significantly affect the performance of BiLSTM models due to their ability to learn temporal sequences and exhibits memory. It did, however, improve the performance of CNNs (see Table 2) which were challenged in distinguishing between the two interval classes (see Figure 6).

Table 2. Comparison of accuracy between 4-class (S1, systole, S2, diastole) and 3-class (S1, systole/diastole, S2) models

Model Type	4-class Model Accuracy	3-class Model Accuracy
fp32 BiLSTM	96.46%	**97.63%**
fp32 CNN	88.98%	**91.97%**

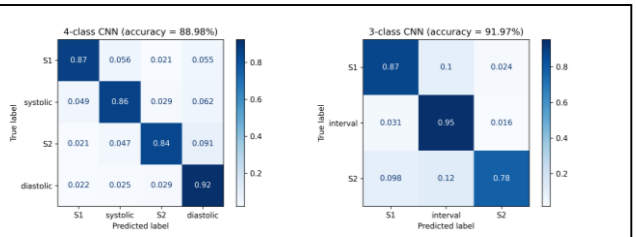

Fig. 6. Confusion matrixes for 4-class and 3-class CNNs

The duration of frames used as inputs for the BiLSTM models was set at 2 seconds (or 2000 samples) for a sampling rate of 1 kHz. This was to ensure that the BiLSTM models learnt the complete sequence of PCG segments which must appear at least once in any 2 second section of a PCG recorded for a patient with a heart rate of above 60 BPM. The upper limit of frame size was set by the dataset due to some recordings having a duration of 2 seconds or less. Recordings shorter than the frame length could not be used to train or test the model. For CNN models' different durations of input frames were considered and the performances of models with different lengths of input frames were compared (see Table 3). It was determined that a frame length of 256 data points corresponding to 2.56 s for a sampling rate of 1 kHz was the optimum value for 3-class and CNNs. 4-class CNNs benefited from a longer frame length of 512 samples or 5.12 s.

Table 3. Comparison of accuracy between CNN models with different input window sizes

Frame Length [samples]	64	128	256	512	1024
3-class 32fp CNN accuracy	88.03%	90.21%	**91.97%**	91.58%	90.26%
4-class 32fp CNN accuracy	71.54%	77.14%	83.64%	**88.98%**	88.36%

The scikit-learn implementation of grid search was used to optimize CNN batch size (see Table 4). The EarlyStopping TensorFlow callback function was used to terminate training if the model overfitted rather than using a set number of epochs.

Table 4. Optimal CNN training hyperparameters determined with grid search

Model	Optimal Batch Size
4-class 32fp CNN	4
3-class 32fp CNN	8

VII. 8-BIT QUANTIZATION

CNNs, in contrast to floating point (fp32) 32-bits BiLSTMs, allowed for 8-bit quantization, which reduced by four times the memory footprint of the model and decreased inference execution duration. Unfortunately, post training quantization (PTQ) reduced too much the accuracy. To minimize that drop, QKeras deep learning framework featuring quantization aware training (QAT), was used to implement an 8-bit CNN. Evaluation results are in Tables 5 and 6. In the future, these results will be improved through the employment of knowledge distillation based QAT, and using a more accurate, even if with a greater model size, fp32 CNN as the teacher.

Table 5. Effect of 8-bit quantization with QAT on model performance

Model Type	Accuracy
fp32 bits CNN	**91.97%**
8-bits CNN	89.84%

The models, shown in Figure 7, were automatically deployed and run on the MCU using the X-CUBE-AI 7.3.0 expansion pack for STM32CubeMX. Inferences were tested on the hardware board featuring the STM32H7 @ 480 MHz.

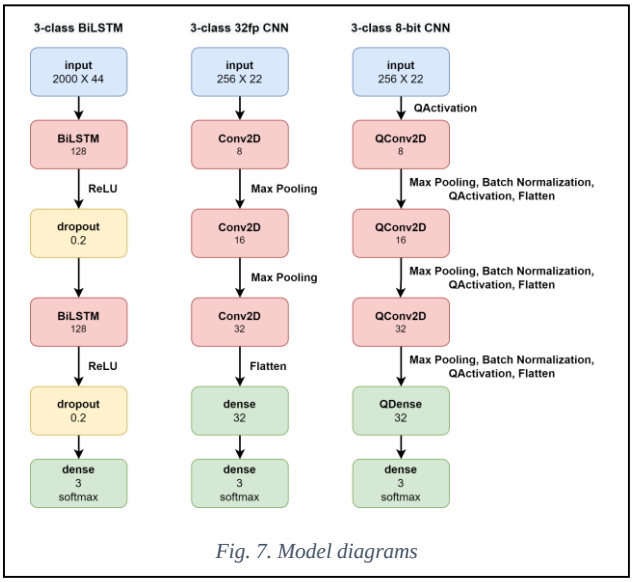

Fig. 7. Model diagrams

Table 6 reports the model footprints in RAM, FLASH, computational complexity measured as number of Multiply and ACCumulate (MACC) operations and single inference time in ms. The latter has been estimated by equation (1).

$$inference\ time = MACC * 9\frac{cycles}{MACC} * \frac{1}{480 * 10^6 \frac{cycles}{sec}} \quad \text{(eq. 1)}$$

9 cycles/MACC is the average number computed by running several benchmarks on the MCU.

Table 5. Resource utilization for different 3-class deep learning models

Model Type	RAM Usage	Flash Usage	Multiply Accumulate (MACC)	Inference Time [ms]
BiLSTM (Keras)	3.91 MiB	2.21 MiB	1,146,464,000	21,496.2*
fp32 CNN (Keras)	44.89 KiB	523.48 KiB	2,310,856	37.751**
8-bit CNN (QKeras)	**45.39 KiB**	**126.3 KiB**	**2,163,912**	**11.994****

*Estimated ~9 cycles/fp32 MACC, STM32H7 – see eq. 1; ** measured on the real hardware MCU

VIII. Post-processing

Due to breaks in continuity of classifications performed by CNN models postprocessing was required to smooth out the segmentation output. A voting algorithm with a sliding window with a length of 9 output samples was utilised for this purpose (see Figure 8). The final classification for each original audio datapoint was set to be that of the most prevalent class in a window of length 9 centred around the datapoint currently being considered.

This process removed single point misclassifications, which were common in the output of CNN models. Such a method of post-processing made the output suitable for use in further stages of the automated stethoscope pipeline (see Figure 8). It improved the total accuracy of segmentation at a very close level of the fp32 CNN (in Table 5) even if the single datapoint misclassifications were constituting a small proportion of total misclassifications (see Table 6.)

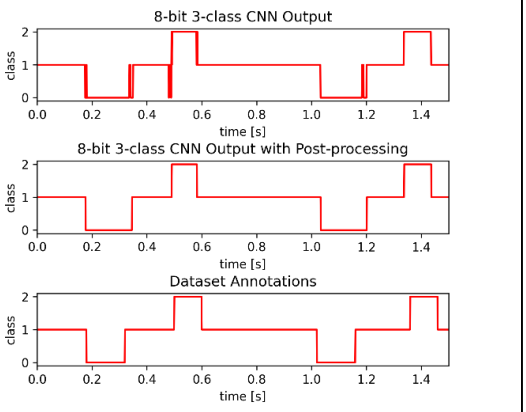

Fig. 8. Effect of post-processing on segmentation output

Table 6. Performance improvements due to post-processing of CNN output

Processing Type	Accuracy
3-class 8-bit CNN	89.84%
3-class 8-bit CNN + post-processing	**91.65%**

IX. Inference Pipeline

The proposed 8-bit CNN model required that a series of pre-processing and post-processing tasks shall be carried on the MCU of an automated stethoscope in addition to inference of the neural network (see Figure 9). The downsampling stage could be integrated with STFT by introducing a larger stride between STFT windows. STFT magnitudes shall be normalized and quantized to 8 bits prior to being inputted into the CNN for inference.

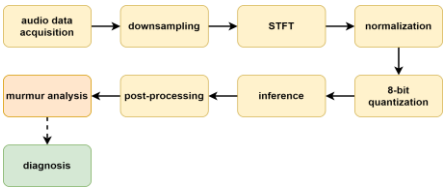

Fig. 9. Complete pipeline that shall be implemented on an MCU for PCG segmentation

X. Conclusions

This work proposed a tiny Keras fp32 and 8-bits QKeras CNN models for PCG segmentation, optimized for inference on low cost ARM Cortex-M MCU and which was comfortably deployed into the available embedded FLASH and RAM of the STM32H7 to enable the development of a tiny consumer-grade automated stethoscope. Even a MCU with lower embedded memory could be used due to the memory footprint of the 8-bits CNN model with QAT quantization.

A 3-class segmentation model was deemed enough to identify S1, S2 and systole/diastole which can then be further analysed to perform the final diagnosis. BiLSTM and CNN models' deployability on STM32 MCUs were evaluated in terms of the memory resource utilization and inference speeds. The 8-bits CNN model was more parsimonious for the target application due to its smaller footprint and faster inference as reported in Table 5. This model was optimized by employing 8-bit QAT (QKeras) quantization. Thanks to the post processing the accuracy drop was reduced to a marginal value with respect to the fp32 CNN.

In the future, knowledge distillation will be used to improve furthermore the 8-bit CNN accuracy. Moreover, the deployment costs on the MCU could be further optimized by investigating hybrid binary weights and activation quantization. Inference time will be improved by introducing a larger stride STFT and by reducing the length of the CNN input frame in samples while maintaining the same length in time. This will come at the cost of decreased temporal resolution and is therefore a compromise that must be carefully evaluated.

References

[1] World Health Organization (2017) Cardiovascular Diseases (CVDs) Fact Sheet. http://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] Renna, Francesco & Oliveira, Jorge & Coimbra, Miguel. (2019). Deep Convolutional Neural Networks for Heart Sound Segmentation. IEEE Journal of Biomedical and Health Informatics. PP. 1-1. 10.1109/JBHI.2019.2894222.

[3] Gaona, A. J., & Arini, P. D. (2020). Deep recurrent learning for heart sounds segmentation based on instantaneous frequency features. *Elektron*, *4*(2), 52–57. https://doi.org/10.37537/rev.elektron.4.2.101.2020

[4] Yin, Yibo & Ma, Kainan & Liu, Ming. (2020). Temporal Convolutional Network Connected with an Anti-Arrhythmia Hidden Semi-Markov Model for Heart Sound Segmentation. Applied Sciences. 10. 7049. 10.3390/app10207049.

[5] Springer, D. (2019). Logistic Regression-HSMM-based Heart Sound Segmentation (version 1.0). *PhysioNet*. https://doi.org/10.13026/vnt9-kf93

[6] Springer, D., Tarassenko, L., & Clifford, G. (2015). Logistic Regression-HSMM-based heart sound segmentation. *IEEE Transactions on Biomedical Engineering*, 1–1. https://doi.org/10.1109/tbme.2015.2475278

[7] J. F. Kaiser, "Digital Filters" - Ch 7 in "Systems analysis by digital computer", Editors: F.F. Kuo and J.F. Kaiser, p 218-285. John Wiley and Sons, New York, (1966).

[8] Schmidt, S. E., Holst-Hansen, C., Graff, C., Toft, E., & Struijk, J. J. (2010). Segmentation of heart sound recordings by a duration-dependent hidden Markov model. *Physiological Measurement*, *31*(4), 513–529. https://doi.org/10.1088/0967-3334/31/4/004