WaveFlow – Towards Integration of Ultrasound Processing with Deep Learning

Piotr Jarosik*, Michal Byra†, Marcin Lewandowski‡§
*Department of Information and Computational Science,
†Department of Ultrasound,
‡Laboratory of Professional Electronics,
Institute of Fundamental Technological Research PAS,
Warsaw, Poland;
§us4us Ltd., Warsaw, Poland;
email: pjarosik@ippt.pan.pl, mbyra@ippt.pan.pl, mlew@ippt.pan.pl

***Abstract*—The ultimate goal of this work is a real-time processing framework for ultrasound image reconstruction augmented with machine learning. To attain this, we have implemented WaveFlow – a set of ultrasound data acquisition and processing tools for TensorFlow. WaveFlow includes: ultrasound Environments (connection points between the input raw ultrasound data source and TensorFlow) and signal processing Operators (ops) library. Raw data can be processed in real-time using algorithms available both in TensorFlow and WaveFlow. Currently, WaveFlow provides ops for B-mode image reconstruction (beamforming), signal processing and quantitative ultrasound. The ops were implemented both for the CPU and GPU, as well as for built-in automated tests and benchmarks. To demonstrate WaveFlow's performance, ultrasound data were acquired from wire and cyst phantoms and elaborated using selected sequences of the ops. We implemented and evaluated: Delay-and-Sum beamformer, synthetic transmit aperture imaging (STAI), plane-wave imaging (PWI), envelope detection algorithm and dynamic range clipping. The benchmarks were executed on the NVidia® Titan X GPU integrated in the USPlatform research scanner (us4us Ltd., Poland). We achieved B-mode image reconstruction frame rates of 55 fps, 17 fps for the STAI and the PWI algorithms, respectively. The results showed the feasibility of real-time ultrasound image reconstruction using WaveFlow operators in the TensorFlow framework. WaveFlow source code can be found at github.com/waveflow-team/waveflow.**

***Index Terms*—beamforming, deep learning, machine learning, tensorflow**

I. Introduction

In recent years, we can observe a growing interest in the use of deep learning methods and GPU processing in ultrasound imaging. Various machine learning tasks have been considered for this modality, eg. classification, segmentation (for CAD systems) and image enhancement [4], [5], [6]. What is more, the implementation of ultrasound imaging pipeline directly on GPU becomes more popular and realistic to perform, mostly due to increasing power of the available technology [7], [9]. The Software Defined Ultrasound paradigm naturally opens the possibility to apply machine learning tools at any stage of the imaging pipeline, especially to process raw ultrasound radio-frequency (RF) data. This idea is highly promising and becomes well supported by hardware, however the appropriate software architecture still has to be developed.

Ultrasound image reconstruction algorithms can be defined and implemented as a sequence of signal processing operations like: filtering, Delay-and-Sum (DAS) beamforming, envelope detection and dynamic range clipping. These operations have a clear functional definition and constitute the data processing pipeline. The target implementation of the pipeline should be easy to rearrange depending on the needs, e.g. the software should provide a convenient way to introduce new signal filtering operations. The idea of ultrasound imaging pipeline has been already considered and evaluated in previous publications [8], [9]. Each of this work considers own, different implementation of the pipeline framework, with operations implemented specifically for one type of the processing device.

In this work we present WaveFlow [1] – a set of ultrasound data acquisition and processing tools for TensorFlow. TensorFlow is a framework, which allows to define a dataflow graph, whose nodes represent operators (units of computation, *ops*), and edges represent ops input/output data (tensors) [2]. Each op can have an implementation for multiple processing devices, like CPU or GPU. This software gained the most popularity, thanks to a broad library of available machine learning algorithms, however the framework can be used not only for AI training and inference. In WaveFlow, we provide a collection of high-level signal processing ops and tools, which can constitute imaging pipeline. Those operators can be added to the graph, and process RF data to reconstruct B-mode frames. Furthermore, WaveFlow and TensorFlow ops can be placed in the same graph what provides a convenient way to augment ultrasound image reconstruction with machine learning.

II. Methodology

WaveFlow includes ultrasound Environments and signal processing Operators (ops) library for TensorFlow.

A. Ultrasound Environment

The Ultrasound Environment is a connection point between the input raw RF data source (i.e. dataset, simulation or hardware scanner) and TensorFlow framework. It feeds the graph with:

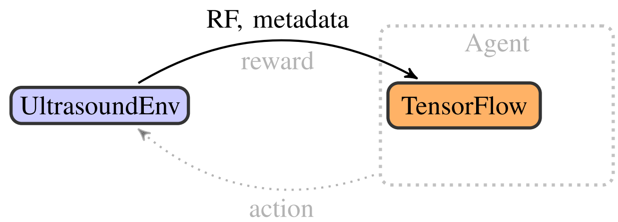

Fig. 1. Ultrasound Environment provides observations (RF data and metadata), which can be processed by graph defined in TensorFlow. Currently, Ultrasound Environment is an *read-only environment* specialized for ultrasound imaging.

- raw RF data stored as a TensorFlow tensor,
- metadata, e.g.:
  - examined physical environment parameters, like speed of sound,
  - ultrasound probe parameters, like aperture size and sampling frequency.

The name *Ultrasound Environment* loosely refers to reinforcement learning task environments, where some software agent takes some actions so as to maximize some cumulative reward [10]. WaveFlow currently provides only observations from the environment, so it is possible to reconstruct ultrasound image on further steps (see figure 1). However, this idea can be extended in future e.g. with actions to perform.

B. Signal Processing Operators

The data provided by the Ultrasound Environment can be processed by operators registered in TensorFlow framework. WaveFlow includes implementations of the following ops:

- pre-processing: e.g. FIR filter;
- mid-processing: Delay-and-Sum (DAS) beamformer, including the Synthetic Transmit Aperture Imaging (STAI) and the Plane Wave Imaging (PWI);
- post-processing: e.g. envelope detection, dynamic adjustment.

These ops can be chained into a classical ultrasound image reconstruction pipeline, as presented in figure 2. The pipeline can be augmented with machine learning tools at any step of processing. Here we present two simple use cases as an example: applying neural network to estimate homodyned K distribution parameters from beamformed data (figure 3), and applying convolutional network to classify beamformed image (figure 4). Our public library of ops will be extended with the quantitative ultrasound (qUS) estimators in future.

Whenever possible, convenient and efficient, we implemented WaveFlow operations as a composition of ops available in the TensorFlow core library (e.g. Hilbert transform using the FFT). For other cases (e.g. DAS) we have provided our own implementations for the CPU and GPU.

III. Evaluation

The most recent performance evaluations are available at [1]. Here we present the imaging pipeline throughput (figure 2), evaluated using the following software and hardware:

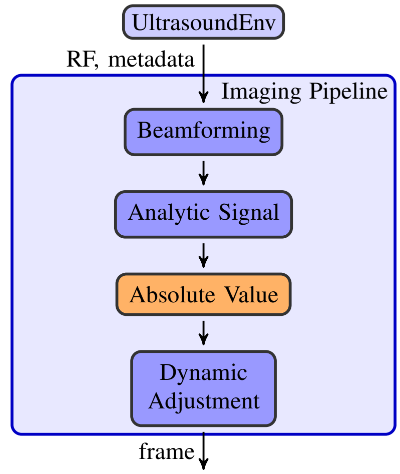

Fig. 2. Pipeline illustrating B-mode image reconstruction. Blue boxes represent operators defined in WaveFlow, orange – TensorFlow.

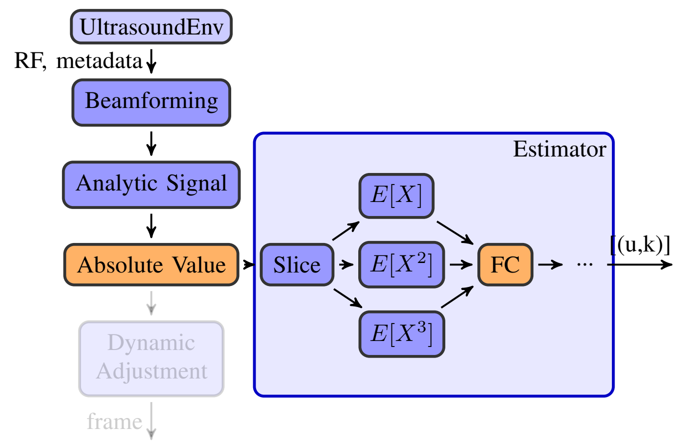

Fig. 3. Example use case: we can estimate the u and k parameters of the Homodyned K distribution, which is widely used to model backscattered echo statistics. The estimation can be performed using fully connected multi-layer neural network. As the result, quantitative parametric maps of distribution's parameters are obtained. Blue boxes represent operators defined in WaveFlow, orange – TensorFlow.

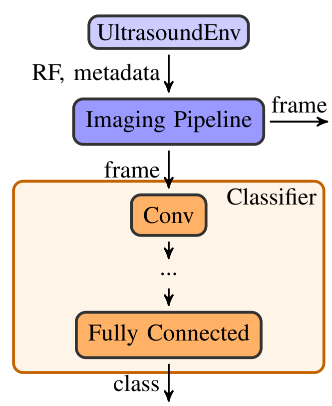

Fig. 4. Example use case: we can classify B-mode frame using convolutional neural network. As the result, we obtain B-mode frame and its class. Blue boxes represent operators defined in WaveFlow, orange – TensorFlow.

TABLE I
RESULTS ACHIEVED ON EACH STEP OF ULTRASOUND IMAGING PIPELINE.

Step	STAI [ms/frame]	PWI [ms/frame]
Beamforming	11	55
Envelope Detection	5	4
Dynamic Adjustment	2	1

- WaveFlow ver. 0.1 (March 28, 2018);
- TensorFlow ver. 1.6.0;
- CUDA ver. 9.0;
- NVidia Titan X GPU integrated in the USPlatform research scanner (us4us Ltd., Poland).

Experiments were performed according to the following assumptions:

- STAI: input tensor with a shape (128, 64, 2048) [number of acquisitions, number of channels, number of samples] per frame, data type: float64, output frame: (2048, 128), single focal point, wire/cyst phantom data examined, available here [3];
- PWI: input tensor with a shape (11, 192, 2048), data type: float64, output frame (512, 128), wire/cyst phantom data examined, available here [3];
- dynamic clipped to range [0, 30] dB.

Table I presents the number of miliseconds required to process an input data frame at each step of the pipeline. In total, 18 ms/frame was required for the STAI (55 FPS), and 60 ms/frame (17 FPS) for the PWI.

IV. CONCLUSION

In this work, we implemented and verified the feasibility and efficiency of WaveFlow – a set of ultrasound data acquisition and processing tools for TensorFlow on a GPU-based ultrasonic research scanner. The results of the experiments show, that our software can be successfully used to reconstruct ultrasound B-mode images in real-time.

Tight integration with TensorFlow opens a possibility to conveniently integrate the imaging pipeline with machine learning algorithms. It also supports execution on multiple processing units, for the price of accepting limitations of this software (e.g. static dataflow graph). WaveFlow extends the TensorFlow framework with a collection of general purpose, signal processing algorithms. It can be used to reconstruct ultrasound images in particular, and can be executed on GPU or CPU, in experimental real-time environment or for research purposes.

Waveflow is an open-source effort (github.com/waveflow-team/waveflow) and contributors are welcome.

REFERENCES

[1] https://github.com/waveflow-team/waveflow/
[2] https://www.tensorflow.org/
[3] https://brain.fuw.edu.pl/edu/index.php/USG
[4] Byra Michał, et al. ”Combining Nakagami imaging and convolutional neural network for breast lesion classification.” *Ultrasonics Symposium (IUS), 2017 IEEE International. IEEE*, 2017.
[5] Smistad Erik, Østvik Andreas. ”2D left ventricle segmentation using deep learning.”, *Ultrasonics Symposium (IUS), 2017 IEEE International. IEEE*, 2017
[6] Perdios Dimitris, et al. ”A deep learning approach to ultrasound image recovery.” *Ultrasonics Symposium (IUS), 2017 IEEE International. IEEE*, 2017.
[7] Lewandowski Marcin. ”Ultrasound Medical Imaging in the GPU Era.” *GPU Technology Conference, San Diego*, 2018.
[8] Lewandowski Marcin, et al. ”Modular & scalable ultrasound platform with GPU processing.” *Ultrasonics Symposium (IUS), 2012 IEEE International. IEEE*, 2012.
[9] Hyun Dongwoon. ”Customizable Ultrasound Imaging in Real-Time Using a GPU-Accelerated Beamformer.” *GPU Technology Conference, San Diego*, 2018.
[10] Szepesvari Csaba. ”Algorithms for reinforcement learning.” *Synthesis lectures on artificial intelligence and machine learning* 4.1 (2010): 1-103.