The Feasibility of Deep Learning Algorithms Integration on a GPU-based Ultrasound Research Scanner

Piotr Jarosik*, Marcin Lewandowski*†
*Laboratory of Professional Electronics,
Institute of Fundamental Technological Research PAS,
Warsaw, Poland;
†us4us Ltd., Warsaw, Poland;
email: pjarosik@ippt.pan.pl, mlew@ippt.pan.pl

***Abstract*—Ultrasound medical diagnostics is a real-time modality based on a doctor's interpretation of images. So far, automated Computer-Aided Diagnostic tools were not widely applied to ultrasound imaging. The emerging methods in Artificial Intelligence, namely deep learning, gave rise to new applications in medical imaging modalities. The work's objective was to show the feasibility of implementing deep learning algorithms directly on a research scanner with GPU software beamforming. We have implemented and evaluated two deep neural network architectures as part of the signal processing pipeline on the ultrasound research platform USPlatform (us4us Ltd., Poland). The USPlatform is equipped with a GPU cluster, enabling full software-based channel data processing as well as the integration of open source Deep Learning frameworks. The first neural model (S-4-2) is a classical convolutional network for one-class classification of baby body parts. We propose a simple 6-layer network for this task. The model was trained and evaluated on a dataset consisting of 786 ultrasound images of a fetal training phantom. The second model (Gu-net) is a fully convolutional neural network for brachial plexus localisation. The model uses 'U-net'-like architecture to compute the overall probability of target detection and the probability mask of possible target locations. The model was trained and evaluated on 5640 ultrasound B-mode frames. Both training and inference were performed on a multi-GPU (Nvidia Titan X) cluster integrated with the platform. As performance metrics we used: accuracy as a percentage of correct answers in classification, dice coefficient for object detection, and mean and std. dev. of a model's response time. The 'S-4-2' model achieved 96% classification accuracy and a response time of 3 ms (334 predictions/s). This simple model makes accurate predictions in a short time. The 'Gu-net' model achieved a 0.64 dice coefficient for object detection and a 76% target's presence classification accuracy with a response time of 15 ms (65 predictions/s). The brachial plexus detection task is more challenging and requires more effort to find the right solution. The results show that deep learning methods can be successfully applied to ultrasound image analysis and integrated on a single advanced research platform.**

I. Introduction

Machine learning, and particularly deep learning, could become a new technology of choice for ultrasound applications, such as guidance and automated diagnostic. The first commercial implementations are emerging. In 2016, Samsung-Medison applied deep learning technology to ultrasound imaging for breast lesion analysis [1]. A software add-on module uses BI-RADS scores for standardized analysis and classification of suspicious lesions and supports decision on benign or malignant lesions. A few start-up companies (Butterfly Network, BayLabs) are working on new applications of Artificial Intelligence in ultrasound.

Our objective was to verify the feasibility of implementing deep learning technology directly on a research ultrasound scanner. As our test platform, we have chosen a research system USPlatform (us4us Ltd., Poland) with GPU-based beamforming. The USPlatform is an all-in-one solution with integrated PC and GPU cluster enabling implementation of advanced signal processing algorithms. The very same GPU resources can be used (shared) for implementation of deep learning algorithms with the help of available open-source frameworks (eg. Theano, Caffe, TensorFlow, Torch).

In this work, we consider two tasks in the context of ultrasound imaging: classification and segmentation of B-mode frames. We have implemented two different neural architectures in both tasks. We have trained and evaluated these models on collections of ultrasound images. We have followed the Occam's razor principle "*entities are not to be multiplied without necessity*", by reducing the number of parameters, and thus the complexity, of today's popular neural network architectures, so that they can better generalize knowledge hidden in the small datasets available.

II. Methodology

In this section, we introduce the ANNs (Artificial Neural Network) architectures and its implementation on the research ultrasound scanner. Because our problem requires both a classification and segmentation of ultrasound images, we have selected ANNs capable of solving both tasks.

The aim of classification it to assign class from a given set C to each available image. We have used three types of layers to construct our simple *S-4-2* neural network architecture for classification:

- fully connected (FC) layers, which represents a linear function of input x,

978-1-5386-3383-0/17/$31.00 ©2017 IEEE

TABLE I
S-4-2 ARCHITECTURE - DETAILS

layer type	description
convolution	16 filters 11×11
max pooling	2×2
convolution	16 filters 7×7
max pooling	2×2
convolution	32 filters 3×3
max pooling	2×2
convolution	64 filters 3×3
max pooling	2×2
fully connected	128 neurons
fully connected	c neurons

- convolutional layers with **filters**, which are convolved with spatial input tensor X,
- max-pooling layers to downsample input tensor X and thus reduce the number of parameters in following layers.

The *S-4-2* model consists of four convolutional (two low-level with 16 filters and two mid-level with 32 filters) and two fully connected layers to increase the receptive field to include the whole input image and produce the output probabilities of classes. A more detailed description is presented in table I. We have compared results achieved by this architecture with the *AlexNet* [2] performance.

The main objective of our second task, segmentation, is to classify not the whole image, but each of its fragments separately. One of the possible solutions to this task is to employ convolutional filters as *local classifiers* of input image regions. This idea leads to a neural network with only convolutional layers, and eventually pooling layers, which is called a Fully Convolutional Network (FCN) [3].

U-net [4] is one of the possible incarnations of FCN. It has been shown experimentally that this architecture achieves very good results in biomedical segmentation problems. This network consists of two parts: *contraction* and *expansive* paths, both with a large number of convolutional filters. The similar number of filters on both paths makes this architecture symmetrical and (when drawn in a particular way) *u-shaped*. To increase the amount of diverse scale information propagated through the expansive path, features from the contraction path are merged with upsampled data. Still, the model is defined by a large quantity of parameters (mostly due to numerous filters in the middle part of the network), making it prone to overfit.

To reduce the number of parameters and improve overall *U-net* performance, we have introduced **inception blocks** in the middle part of the network and named this architecture ***Gu-net***. Inception blocks were first proposed in the *GoogLeNet* model, which was developed by Szegedy et. al [5]. The main idea behind this structure is to split one layer of filters with the same kernel size into multiple layers with kernels of different sizes and one max-pooling layer. One is not restricted to any specific partition; in our work we have evaluated:

- $3 \times 3 \rightarrow (1 \times 1, 3 \times 3, 5 \times 5)$ split, what gives in a result original inception block from [5] (*std. inception block*)

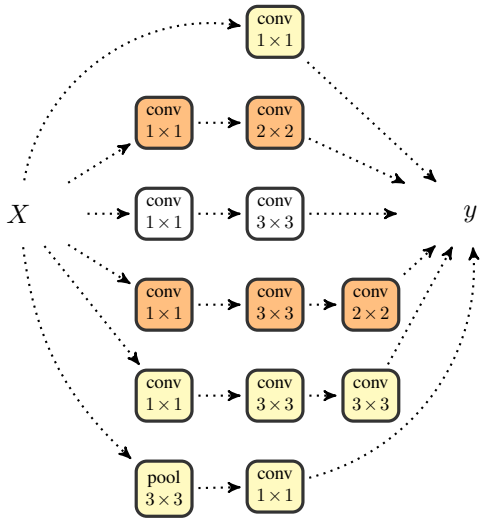

Fig. 1. Inception block after applying tensor compression and kernel factorization.

- $(1 \times 1, 3 \times 3, 5 \times 5) \rightarrow (1 \times 1, 2 \times 2, 3 \times 3, 4 \times 4, 5 \times 5)$ split to *increase* models expressiveness (*dense inception block*).

We have compared both structures to check whether the second split improves network performance.

One must note that the **split** operation may still produce models with large number of parameters, mostly due to the introduction of kernels with size greater than the original (i.e. $3 \times 3 \rightarrow 5 \times 5$). To reduce the scale of this issue, we can **compress** tensors along channel dimension. To achieve this, we prepended the inception block components with a small number of 1×1 filters (fig. 1). Moreover, convolution with a 5×5 kernel can be replaced by two successive convolutions with 3×3 kernels. The kernel **factorization**, proposed in [6], reduces the number of parameters from 25 to 18 for a single filter. One can go further and replace all 3×3 kernels with $3 \times 1 \rightarrow 1 \times 3$ filters. However, for the sake of simplicity, we limited ourselves to the first type of factorization in the evaluation procedure.

We have made further modifications to the *U-net* and *Gu-net* architectures, so that they are better suited to the challenge awaiting them in the evaluation process. In the rest of the work, we focus on the segmentation task with only one target class ($c = 1$); in this case the model's result can be treated as an object detection mask. In order to evaluate the probability that the object is present in the input image, and to prepare the model for images without the target, we have introduced an additional binary classifier by extending the contraction path with an additional fully connected layer. We have applied this modification to both neural architectures.

We have prepared and implemented three versions of the *Gu-net* architecture. The first one (v1) contains standard $(1 \times 1, 3 \times 3, 5 \times 5)$ inception blocks. The second (v2) and third one (v3) use dense inception blocks, which only differ in the number of base filters. A detailed architecture of the

978-1-5386-3383-0/17/$31.00 ©2017 IEEE

TABLE II
GU-NET ARCHITECTURE (DETAILS)

layer type	Gu-net v1	Gu-net v2	Gu-net v3
convolution	32 filters 7×7	same as v1	same as v1
max pooling	2×2	same as v1	same as v1
convolution	64 filters 3×3	same as v1	same as v1
max pooling	2×2	same as v1	same as v1
inception	std. L_5 = 128	dense L_5 = 128	dense, L_5 = 64
max pooling	2×2	same as v1	same as v1
inception	std. L_7 = 256	dense L_7 = 256	dense L_7 = 128
max pooling	2×2	same as v1	same as v1
inception	std. L_9 = 512	dense L_9 = 512	dense L_9 = 256
upsampling	2×2	same as v1	same as v1
inception	std. L_{11} = 256	dense L_{11} = 256	dense L_{11} = 128
upsampling	2×2	same as v1	same as v1
inception	std. L_{13} = 128	dense L_{13} = 128	dense L_{13} = 64
upsampling	2×2	same as v1	same as v1
convolution	64 filters 3×3	same as v1	same as v1
upsampling	2×2	same as v1	same as v1
convolution	32 filters 3×3	same as v1	same as v1
convolution	c filters 1×1	same as v1	same as v1

Gu-net is provided in table II.

III. EXPERIMENTS

We have evaluated neural networks on two different datasets, for two tasks: classification and image segmentation of ultrasound frames. The objective of our work was to evaluate the performance of the described models and to show the feasibility of implementing deep learning algorithms directly on a research scanner with GPU software beamforming. In this section, we describe the experimental environment, which datasets we used and how we assessed the models' performance.

A. Evaluation

We have used the k-fold cross validation method to evaluate the neural networks presented. In this procedure, we divide a given dataset D into k separated subsets $D_1, D_2, ...D_k$, $\forall_{i \neq j} D_j \cap D_i = \emptyset$. Next, we do k iterations: in iteration i we train the neural network on $D - D_i$ and evaluate its performance using D_i. The main advantage of cross validation is that each sample from D is used solely to evaluate our model.

We have performed our experiments on a 4-GPU cluster integrated with our platform. Each GPU is a NVIDIA GTX Titan X card equipped with 12 GB GDDR5 memory. We have implemented neural networks using Tensorflow [9] with CUDA support.

We have evaluated our classification models on a dataset containing 786 ultrasound frames of a baby's phantom [7]. Each frame presents one of three body parts: abdominal, head or legs. To increase the diversity of our samples, we have produced an augmented dataset: we drew a sample from the original collection and applied to it a random transformation.

TABLE III
CLASSIFICATION NETWORKS RESULTS

model	accuracy	$\mathbf{t_c}$	pred/s	total params
S-4-2	**0.96**	**3 ms**	**334**	**0.176 mln**
AlexNet	0.95	6 ms	171	57 mln

TABLE IV
SEGMENTATION NETWORKS RESULTS

model	d	accuracy	$\mathbf{t_s}$	pred/s	total params
U-net	0.5554	0.76	**5 ms**	**186**	7.8 mln
Gu-net v1	0.6333	0.76	13 ms	76	1.3 mln
Gu-net v2	0.6310	0.75	18 ms	54	1 mln
Gu-net v3	**0.6390**	**0.76**	15 ms	65	**0.357 mln**

The transformation is composed of a rotation by an angle value from uniform distribution $Uniform[-180, 180]$, a possible horizontal/vertical flip and a width/height shift of the image. We repeated this procedure until we created a dataset of a fixed size (2000 samples). As a result, the model became *prepared* for this sort of possible linear transformations. To assess the performance of the model, we have used a 5-fold cross validation and measured:

- top-1 classification accuracy:

$$accuracy = \frac{|V|_f}{|V|}$$

  where $|V|_f$ is a number of properly classified samples by f, and $|V|$ is the total number of samples;
- inference time t_c for a single image.

Models for segmentation can be evaluated in an analogous way to the one shown above. We have used a dataset containing 5640 B-mode frames [8]. Most of them show the brachial plexus, which is our Object of Interest (*OoI*); each such frame has a brachial plexus presence mask attached. There are two tasks: the binary classification of OoI's presence and image segmentation. We have evaluated the neural networks described using a 6-fold cross validation. We measured:

- presence classification $accuracy$,
- Dice coefficient between predicted probabilities and a ground truth mask in vector notation:

$$d = \frac{2Y_{pred} \cdot Y}{\sum Y_{pred,i} + \sum Y_i}$$

- segmentation time of a single image t_s.

B. Results

Table III presents the evaluation results of neural networks in the classification task: *S-4-2*, which was introduced in our work, and the *AlexNet*, popular in literature. Both networks achieved similar top-1 classification accuracy; *S-4-2* achieves slightly better results and requires fewer training epochs. The advantages of *S-4-2* may be due to its smaller number of parameters (0.176 vs. 57 mln), which makes it less prone to overfit. Both neural network implementations are very

978-1-5386-3383-0/17/$31.00 ©2017 IEEE

Fig. 2. Ultrasound B-mode image presenting a baby phantom part with classification results.

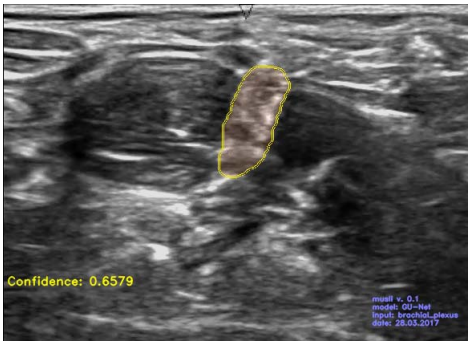

Fig. 3. Ultrasound B-mode image presenting the brachial plexus with its presence mask.

responsive (171 and 334 predictions/sec); however, *S-4-2* was twice as fast as *AlexNet* because of fewer component layers. In figure 2, we present an example frame of a baby's body part with classification results.

In table IV, one can find the evaluation results of neural networks in the segmentation task. Although *Gu-net* and *U-net* have similar presence classification accuracy, the first one achieved a much higher Dice score. All versions of *Gu-net* perform comparably in the segmentation task. The model with *dense* inception blocks and a reduced number of parameters (v3) achieved a slightly better result than others. This version of the model is also defined by the smallest number of parameters. However, this characteristic does not make *Gu-net v3* the fastest – it is the *U-net* network which has the shortest inference time. The reason for this is the strong dependence of network response time on the number of component layers. *U-net* may have more parameters, but it also has fewer layers than the *Gu-net* equipped with inception blocks. In figure 3, we a present example frame with the segmentation mask of the brachial plexus.

IV. Conclusion

In this work, we have implemented and verified the feasibility and efficiency of deep learning algorithms in the classification and segmentation of B-mode frames on a GPU-based ultrasonic research scanner. The results of experiments show that deep artificial neural networks can be successfully used in these tasks.

The best score was achieved by low dimensional neural models (with a small number of weights). These networks seem to be better at generalizing the knowledge hidden in small medical datasets. In this case, *the simpler the model is, the better results it achieves*. It is possible to increase the complexity of the architecture when more data is available, as artificial neural networks are flexible enough structures to adapt to the problem at hand.

References

[1] 'Deep Learning' Technology applied to Diagnostic Ultrasound Imaging, DI Europe. OCTOBER 2016.
[2] A. Krizhevsky, I. Sutskever, G.E. Hinton. "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems 25* ed. F. Pereira and C. J. C. Burges and L. Bottou and K. Q. Weinberger, pages 1097–1105, 2012.
[3] J. Long, E. Shelhamer, T. Darrell, "Fully convolutional networks for semantic segmentation", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.*, 2015.
[4] O. Ronneberger, P. Fischer, T. Brox, "U-net: Convolutional networks for biomedical image segmentation", *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Springer, LNCS, Vol.9351:234–241, 2015.
[5] Ch. Szegedy, et al., "Going deeper with convolutions", *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.
[6] Ch. Szegedy, et al., "Rethinking the inception architecture for computer vision", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
[7] C. Cortes, et al., "Ultrasound Image Dataset for Image Analysis Algorithms Evaluation", *Innovation in Medicine and Healthcare 2015*, Springer, Cham, pages 447–457, 2016.
[8] https://www.kaggle.com/c/ultrasound-nerve-segmentation
[9] https://www.tensorflow.org/

978-1-5386-3383-0/17/$31.00 ©2017 IEEE