International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 4 Issue: 1
ISSN: 2454-4248
69 – 73

Enhancing Effeciency of Ejection Fraction Calculationin the Left Ventricle

Nesrin Abubakr Kamal-Eldeen
Department of Computer Science
Arab Academy for Science,
Technology &
Maritime Transport
Cairo, Egypt
Email: noon.ab@gmail.com

Alaa Eldeen Hamouda
Faculty of Computer Science
Cairo Egypt
Al-Azhar university
Email: dr.alaa.hamouda@gmail.com

Mostafa Abd El Azim
Department of Computer Science
Arab Academy for Science,
Technology &
Maritime Transport
Cairo, Egypt
Email: melbaqary@gmail.com

Abstract—The calculation of the cardiac ejection fraction is important for determining whether or not a patient suffers from cardiovascular disease. However, manual calculation of the ejection fraction (EF) is prone to errors and is known to be prohibitively time-consuming. As such, there have been endeavors to automate this process for the sake of saving time as well as improving accuracy of estimation.

Recently,GPUhave been proposed to enhance the performance of machine learning algorithms that attempt to estimate the EF. In addition, these algorithms are considered a necessary component in solving computational efficiency issuesencountered in dealing with hugeDigital Imaging and Communications in Medicine (DICOM)datasets.

In this study, we useda DICOM dataset of cardiac magnetic resonance imaging for 1200 human cases with different ages and gender to calculate the ejection fraction in the left ventricle.Convolutional Neural Network (CNN) was the selected neural network for the training phase of segmenting the LV and volume calculation.

Our target is enhancing efficiencyof CNN to speedup training phase, and subsequently the prediction of the CVDs by experimenting with different GPU-based parallelism techniques, namely Data Parallelism (DP)and Model Parallelism (MP) in addition to the generic use of multiple GPUs. Specifically, we performed four variants of experiments; the first was using GPUs without applying any control on its behavior, the second two variants involve experiments using either DP alone or MP alone on multiple GPUs, while the fourth and final variant involves combining both DP and MP. This was done on Amazon EC2 instances that support up to 8 GPUs per instance. We used two EC2 instances to apply our experiment on 16 GPUs. Our experiments show that our proposed combination of both DP and MP havethe bestcomputational efficiency. Precisely, a speedup of up to 9.88 (over a single GPU) was achieved when using 16 GPUs in parallel with combined DP and MP.

Keywords- *Multiple GPUs; Data parallelism; Model Parallelism;Convolutional Neural Network*

******

I.INTRODUCTION

Cardiovascular disease (CVD)isconsidered the most common cause of death all over the world [11,13]. Cardiologists depend on MRI to predictdeclined cardiac functions by calculatingthe cardiac ejection fraction (EF)manually [2].

Calculation of EF depends on medical imaging of the cardiac muscle. Cardiac catheterization is an invasive method to show contrast between cardiac muscle wall and the blood flow by inserting a catheter from arm or leg through the artery till reach the heart, during catheterization images used for calculating the EF is taken. There are several types of imaging methods including echocardiography, magnetic resonance imaging (MRI), Computed Tomography (CT), that aren't invasive and doesn't need injecting patient with any chemical to define the LV borders. Nuclear Cardiac Imaging required injecting patient with radioactive material and need special types of cameras to take the images. Each imaging type then is used in manually, semi-automated, or automated technique to calculate the volume of blood at the start and end of each beat [7]. For example, Simpson, area-length, and ellipsoid methods [3]. Today, after the development of machine learning algorithms, there is now a better solution that automates the whole cycle of calculation in addition to prediction of any CVDs.

Various datasets are availablefor researchers to build better automated techniques with higher accuracy and efficiency.Sunnybrook Cardiac Data (SCD) [15, 16] was published ina competition to detect the pathology of affecting LV of cardiac patients. This dataset included 45 cases composed of short-axis (SAX) MRI. Another dataset was also published in another competition on Kaggle aiming automating calculation of EF on LV [12]**.**

Given an input consisting of long time consumed to train the CNN on the DICOM, the goal is to optimizing the computational efficiency of training the convolutional neural networks (CNN) using multiple GPUs and applying different models of parallelism techniques.

The following is a discussion for relevant topics we used in or research:

A. Ejection Fraction (EF)

Ejection Fraction (EF) is the amount of blood ejected from the left ventricle (LV) with each heart beat in relation to the amount of blood filled the LV during relaxation of cardiac muscle.The EF is traditionally computed as

 ISSN: 2454-4248
69 – 73

$$EF\% = 100 \ * \ \frac{\text{EDV} - \text{ESV}}{\text{EDV}} \qquad (1)$$

Equation (1), EF calculation.

where end diastolic volume (EDV) represents the volume of blood in the LV at the end of diastole when the cardiac muscle is completely relaxed and LV is filled maximally with blood, and end systolic volume (ESV) represents the volume of blood in the LV at the end of systole when the cardiac muscle is maximally contracted and LV pumped the blood out [6].

B. Convolutional Neural Network (CNN)

CNN is a type of feedforward neural networkalgorithmthat has been shown to display high accuracy in dealing with the medical image processing [1,8,18], characterized by high learning speed [4]. It begins with an input layer containing any number of images followed by a number of hidden layers each consisting of two main steps:

i. The first step is *convolution*which applies a filter containing the required shape that we are searching for inside the image, with this filter being of a smaller size than the image size. For example, assuming the required shape is a horizontal line, this step passes the filter all over the image to find all the horizontal lines inside it, each success enters as an input image into the second step.
ii. The second step is Subsampling or Maxpool which is responsible for storing the resized images from the previous convolution step containing the required object.

The number of filters used in first step is equal to number of convolutions. The two steps may be repeated several times according to the requirement. The output of the last hidden layer (containing the last subsampled images) will be the input for the fully connected layers that represent our classifier. The output layer can consist of one or more items depending on the requirement.For example, if the requirement is discriminating different diseases depending on the shape, there will be number of outputs dependent on the number of diseases we train the network on [5, 8].In our case, we need only one output image distinguishing the LV location.

C. Parallelism in Image Processing

Saxena et al. [17] worked on a detailed survey related to parallelism in image processing. The survey included GPU usage which uses lower power than CPU despite having up to 240 cores which is 30-60 times number of cores used in CPU of servers, in addition to the thread manager that can support more than 10 thousand of threads per each core, can be managed programmatically using several high-level programming languages, but higher cost than other methods. CUDA (Computed Unified Device Architecture) libraries can be managed programmatically using several high-level programming languages, speedy integration with GPU, its main limitation is that integrates only with NVIDIA. Open Computing Language (OpenCL) is less performing than CUDA. Hadoop is less performing than Java due to it its dependency on Matlab in image processing tasks as Matlab is already built on Java. OpenCV is specific to image processing with embedded functions that need extra work from the programmer to master. Java has no cost on exchanging parameters and communication between threads but there is difficulty of debugging an error in multithreaded programs, competing for machine resources affect performance, hardware support for multithreading needs to be provided by the operating system too to reach maximum efficiency.

D. Related Work

Avendi et al. [2] automated segmentation of LV and calculation of EF using MICCAI 2009 challenge dataset and CNN and stacked autoencoders techniques, Accuracy reached percentage up to +/-1.96 SD for EDV, ESV, EF is about 2.4% from manually calculated, Time cost for training phase is 3.4 hours for CNN, 34.25 minutes for Autoencoders phase for 1350 images x 45 groups, applying the model on new image cost 0.25 seconds CNN, 0.002 seconds stacked-AE and 0.2 seconds segmentation. Margeta [10] segmented LV using decision forests was less accurate than [1] due to including papillary muscles and trabeculations that caused over segmentation. Zhen et al. [18] worked on recognition and calculation of EF in LV, RV without segmentation of each chamber alone in MRI images, the best correlation coefficient value was 0.921 and least LV estimation error was 0.010 ± 0.011 using combination of CNN and deep belief nets for recognition.

Kim et al. [9] enhanced performance using multiple GPUs to train CNN based on different frameworks, theano, Caffe, Torch, TensorFlow and CNTK, Maximum speedup was 2.6 using 4 GPU on CNTK 1bit-SGD

II.OVERVIEW OF THE PROPOSED MODEL

We propose a fully automated CNN method to recognize LV on datasetthat includes 1200 real cases of a human MRI with 2D cine DICOM images, each case has 13 different planes, and each plane include 30 slices, representing a cycle of a complete heartbeat while the patientis holding his/her breath, dataset used from SecondAnnualDataScienceBowl[12], including different genders and ages.

For segmentation phase, 110 SAX imageslabeled by hand. Segmentation continued for all slicesusing CNN resulted in a [0,1] range pixels image of same size of input image where 1 represent LV pixel and 0 is out the LV. Table 1 define the CNN layers used in segmentation, where b = batch size, Conv = convolution, BN = batch normalization, ReLU(x) = max(0, x), Sigmoid(x) = 1/(1+exp(x)), this CNN model was used by the winners of Kaggle competition to train neural network [12].Z- score based normalization was applied as in (2) after each convolution layer [14].

TABLE1. CNN LAYERS

Layer	factor	Filter Size	Output Shape
Input			(b, 1, 246, 246)
Conv+BN+ReLU	8	7	(b, 8, 240, 240)
Conv+BN+ReLU	16	3	(b,16,238,238)
MaxPool	2		(b,16,119,119)
Conv+BN+ReLU	32	3	(b,32,117,117)
MaxPool	2		(b,32,58,58)
Conv+BN+ReLU	64	3	(b,64,56,56)
MaxPool	2		(b,64,28,28)
Conv+BN+ReLU	64	3	(b,64,26,26)

 ISSN: 2454-4248
69 – 73

Conv+BN+ReLU	64	3	(b,64,28,28)
Upscale	2		(b,64,56,56)
Conv+BN+ReLU	64	3	(b,64,58,58)
Upscale	2		(b,64,116,116)
Conv+BN+ReLU	32	7	(b,32,122,122)
Upscale	2		(b,32,244,244)
Conv+BN+ReLU	16	3	(b,16,246,246)
Conv+BN+ReLU	8	7	(b,8,240,240)
Conv+sigmoid	1	7	(b,1,246,246)

$$z = (x - \mu) / \sigma (2)$$

Equation (2), Z score formula

Figure1 shows a block diagram illustratingthe training phase, targeting segmenting the region of intertest (ROI).The First step is augmenting the DICOMs by rotation, transposition and scaling, followed by dividing the input batches for processing through the PCIe switch that sends the batches to the GPUs. Then output classifier is the input for the EF calculation phase. Equation (3)is applied on DICOM slices to get the volume of the LV at the start of the heart beat when the heart is maximally contracted representing ESV and at the end of the beat when heart is fully relaxed representing EDV.Both volumes are used in (1) to compute the EF [4]. Figure2 represents a sample of DICOM slicesfrom the used dataset before and after segmentation.

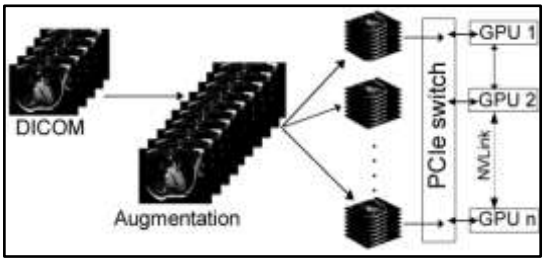

Fig. 1, Training Phase Block.

$$V_t = \sum_{s=}^{i} \frac{x(A_{s,t} + A_{s+1,t})}{2(L_{s+1} + L_s)} + A_{1,t} * \frac{h}{2} + A_{N,t} * \frac{h}{2} \quad (3)$$

Equation (3), LV volume detection from DICOM.A represent the area of slice s at time t,ls is the slice location and w is the slice thickness.

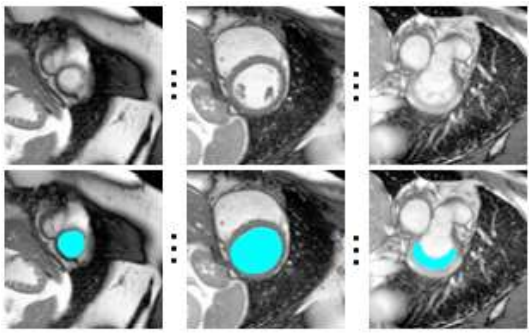

Fig. 2, Image represents3 slices of same case with LV segementation result in the second row represented with the blue color.

The platform used is Amazon EC^2 with NVIDIA Tesla V100 GPUs, on Ubuntu 16 OS, 128 GiB, CUDA 8, NVLink 300 GBps, Network bandwidth 25 Gbps, RAM 488 GiB, PCI-Express (PCIe) fabric switch. The maximum number of GPUs on the EC^2 instance is 8, so on repeating experiment steps with the 16 GPUs, we use 2 EC^2 instances working together in parallel.

The first experiment usesa single GPU and documents the time and accuracy. This is followed by using the different models of parallelism techniques and comparing of the speed up at each phase of the experiment.

Parallelism in our experiment is implemented as follows:

- Generic form of multiple GPUs, without applying any control from programming side on the GPUs or the data batches passed to the GPU.
- Data parallelism (DP) represented by passing separate minibatches over multiple GPUs. Figure 3 illustrates how batches are passed separately on using 2 GPUs, each GPU work on its part of data then the gradient exchange is done through the PCIe that carries the responsibility summing up gradients before updating weights over both GPUs.No direct exchange of weights between GPUs which is expected to be the reason of delay in this technique.DP steps are repeated using 4, 8 and 16 GPUs. Data batches are divided over number of GPUs then PCIe updates the weights for all.Each GPU call the whole CNN model over the batch.

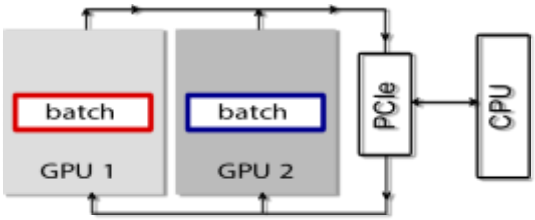

Fig. 3, DP technique architecture using 2 GPUs.

- MP represented by using multiple GPUs, with CNN implemented divided between nodes, same batch pass from node to the nextsubsequently.It exchanges the weights directly with each other using the NVLink without returning to the PCIe.On using 4 GPUs,CNN layers were distributed between the 4 nodes which may be the cause of some delay due batch transfer from node to node. The experiment was repeated with distributing CNN nodes.
- Combining both DP and MP techniques in same operation. This model was done using 4 GPUs, each 2 nodes are considered an MP unit where the CNN model is divided on both nodes, the unit is cloned on the remaining 2 nodes, batches are divided between both MP units which represent the DP part of the technique. The process was repeated using 2 MP units each consists of 4 GPUs. The last turn, as each EC2 instance has maximum 8 GPUs, we cloned MP model on each instance, then we divided the batches between the 2 instances.

III.EXPERIMENTAL RESULTS

Applying the proposed model on training and calculating efficiency on the DICOM including the 1200 case, each case includes 13 different planes with 30 slices for each plan, MRI of human heart, dataset is from [12].Human with different ages and gender to predict the probability of CVDs

 ISSN: 2454-4248
69 – 73

by calculating the EF in LV.Table 2, demonstrates the speedup results based on (4)that compares the execution time using a single GPU (T_s) to time taken by multiple GPUs (T_p) in every step of the experiment, the speedup calculations are represented in the Figure4 chart showing highestefficiency using the generic model of GPUs while least effect on efficiency is by using DP model.

$$S_p = \frac{T_s}{T_p} \quad (4)$$

Equation (4), Speedup.

TABLE 2. SPEEDUP IN RELATION TO 1 GPU

No. of GPU	GPUs Only Speedup	DP Speedup	MP Speedup	Both Speedup
2	1.55	1.18	1.49	-
4	2.63	1.04	1.88	3.04
8	3.43	2.63	2.82	4.65
16	7.18	4.94	-	9.88

Results approves 9.88 times enhancements of efficiency on using 16 GPUs and modelling parallelism techniques using both the MP/DP together, however using the generalized form of GPUs parallelism shows enhancement by 7.18 times with the same number of GPUs, least speedup ratio was by using one of the two techniques DP and MP alone which maximally reached 4.94, DP model is least affected.

As compared with [10] that reached maximum speedup 2.6 using 4 GPU, the speedup using our combined DP/MP technique on 4 GPUs was 3.04 which is higher and on generic GPU model was 2.63, on the other hand using the MP alone or DP alone has less speedup value than 2.

Expected that the cause of delay in DP model is the dependency on the PCIe and CPU to propagate the gradients while on letting the GPU to send the results directly to each other using the NVLink accelerate the training phase in MP and combined form.Also delay in MP alone is expected to be due to single thread for all dataset batches. While the combined form a major breakthrough was achieved as data was divided between the MP units.

DP and MP were shown to be less effective in improving computational efficiency than generic form, while combining them turned out to be even more effective.The two techniques fortunately complement one another with minimal conflict.Using generic form of GPU without interfering in their behavior with any technique proved to have better effect than using DP or MP alone.

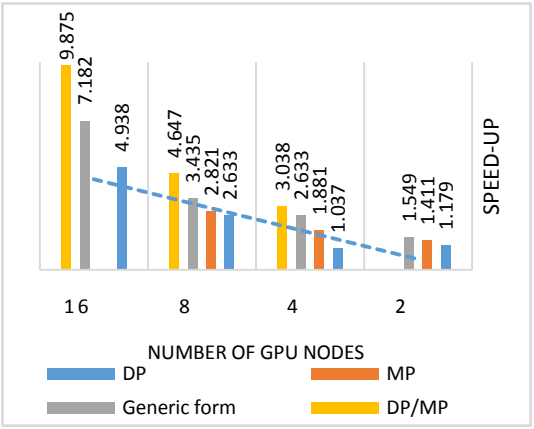

Fig. 4, Demonstration for the speedup ratios in relation to GPUs number

IV. CONCLUSION

DICOM dataset with 1200 cases used for LV volume calculation during the heart beat at its most relaxed state EDV and most contracted state ESV, followed by calculation of the EF which facilitate the prognosis pf CVDs.

Efficiency in CNN is proportional with the number of GPUs irrespective with the used technique. We proposed Generic GPUs, DP, MP, and Combines DP/MP models.

Improving CNN training phase to led to a speedup of 9.88and has improved computational efficiency of EF calculation.

Efficiency of the training phase using the CNN was improves up to 9.88 times using up to 16 GPUs, based on combined DP/MP techniques.

Using multiple GPUs enhance the training performance without affecting the accuracy in an approximately linear relation with the number of GPUs, however on using different types of GPUs causes some truncation/round-off errors.

Additional work will be required to compare the CPUs with the GPUs, keeping same efficiency and speedup results in mind. Expected that usage of CPUs will avoid truncation/round-off errors but cost of keeping same promising speedup will be calculated.

REFERENCES

[1] Avendi, M. R., A. Kheradvar, and Hamid Jafarkhani. "A Combined Deep-Learning and Deformable-Model Approach to Fully Automatic Segmentation of the Left Ventricle in Cardiac MRI." arXiv preprint arXiv:1512.07951 (2015).

[2] Biering-Sorensen T, Shah S, Ananda I, Sweitzer N, Claggett B, Pitt B, Pfeffer M, Solomon S, Shah A. PROGNOSTIC IMPORTANCE OF LEFT VENTRICULAR MECHANICAL DYSSYNCHRONY IN HEART FAILURE WITH PRESERVED EJECTION FRACTION. Journal of the American College of Cardiology. 2016 Apr 5;67(13_S):1484-.

[3] Canciello G, de Simone G, Izzo R, Giamundo A, Pacelli F, Mancusi C, Galderisi M, Trimarco B, Losi MA. Validation of left atrial volume estimation by left atrial diameter from the parasternal long-axis view. Journal of the American Society of Echocardiography. 2017 Mar 31;30(3):262-9.

 ISSN: 2454-4248
69 – 73

[4] Glauner PO. Comparison of Training Methods for Deep Neural Networks. arXiv preprint arXiv:1504.06825. 2015 Apr 26.
[5] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G. Recent Advances in Convolutional Neural Networks. arXiv preprint arXiv:1512.07108. 2015 Dec 22.
[6] Hall JE. Guyton and Hall textbook of medical physiology. Elsevier Health Sciences; 2015 May 18.
[7] Herring W. Learning radiology: Recognizing the basics. Elsevier Health Sciences; 2007 Jun 20.
[8] Highlander T. Efficient Training of Small Kernel Convolutional Neural Networks using Fast Fourier Transform (Doctoral dissertation, Wright State University), 2015.
[9] Kim H, Nam H, Jung W, Lee J. Performance analysis of CNN frameworks for GPUs. InPerformance Analysis of Systems and Software (ISPASS), 2017 IEEE International Symposium on 2017 Apr 24 (pp. 55-64). IEEE.
[10] Margeta, J., 2015. Machine Learning for Simplifying the Use of Cardiac Image Databases (Doctoral dissertation, MINES ParisTech).
[11] Mendis S, Puska P, Norrving B. Global atlas on cardiovascular disease prevention and control. World Health Organization; 2011.
[12] NHLBI, Data Science Bowl Cardiac Challenge Data, December 14, 2015https://www.kaggle.com/c/second-annual-data-science-bowl/data
[13] Nichols M, Townsend N, Scarborough P, Rayner M. Cardiovascular disease in Europe 2014: epidemiological update. European heart journal. 2014 Aug 12:ehu299.
[14] Patro S, Sahu KK. Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462. 2015 Mar 19.
[15] Poudel RP, Lamata P, Montana G. Recurrent fully convolutional neural networks for multi-slice mri cardiac segmentation. InInternational Workshop on Reconstruction and Analysis of Moving Body Organs 2016 Oct 17 (pp. 83-94). Springer, Cham.
[16] Radau P, Lu Y, Connelly K, Paul G, Dick AJ, Wright GA. "Evaluation Framework for Algorithms Segmenting Short Axis Cardiac MRI." The MIDAS Journal – Cardiac MR Left Ventricle Segmentation Challenge, http://hdl.handle.net/10380/3070
[17] Saxena S, Sharma S, Sharma N. Parallel Image Processing Techniques, Benefits and Limitations. Research Journal of Applied Sciences, Engineering and Technology. 2016 Jan 20;12(2):223-38.
[18] Zhen X, Wang Z, Islam A, Bhaduri M, Chan I, Li S. Multi-scale deep networks and regression forests for direct bi-ventricular volume estimation. Medical Image Analysis. 2015 Jul 26