

## ACCELERATING THE DESIGN OF PROBABILISTIC NEURAL NETWORKS FOR COMPUTER AIDED DIAGNOSIS IN MAMMOGRAPHY, EMPLOYING GRAPHICS PROCESSING UNITS

Konstantinos Sidiropoulos<sup>1</sup>, Dionisis Cavouras<sup>2</sup>, Nikolaos Pagonis<sup>3</sup>, Nikos Dimitropoulos<sup>4</sup> and John Stonham<sup>1</sup>

<sup>1</sup> School of Engineering and Design, Brunel University West London, Uxbridge, Middlesex, UB8 3PH, UK, e-mail: Konstantinos.Sidiropoulos@brunel.ac.uk

<sup>2</sup> Medical Image & Signal Processing Laboratory, Department of Medical Instrumentation Technology, Technological Educational Institution of Athens, 12210, Athens, Greece.

<sup>3</sup> Medical Image Processing and Analysis Group (MIPA), Laboratory of Medical Physics, School of Medicine, University of Patras, 26500 Rio, Greece.

<sup>4</sup> Medical Imaging Department, EUROMEDICA Medical Center, Athens, Greece.

**Keywords:** probabilistic neural networks, graphics processing units.

**Abstract.** *The aim of this study is to propose a Probabilistic Neural Network (PNN) classifier system that can operate on a consumer-level graphics processing unit (GPU) and thus, harvest its tremendous parallel computation potential in order to accelerate the training phase. Therefore, the computationally intensive training of a PNN classifier system incorporating the exhaustive search of feature combinations and the leave-one-out techniques, was effectively ported on a medium class GPU device. Programming of the GPU was accomplished by means of the CUDA framework. The proposed system was tested on a real training dataset comprising 80 patterns, each consisting of 20 textural features extracted from digital mammograms (40 normal and 40 containing micro-calcifications) by an experienced physician. The developed GPU-based classifier was trained and the required time was measured. The latter was then compared with the respective training time of the same classifier running on a typical CPU and programmed in the C programming language. According to experimental results, the proposed GPU-based classifier achieved significantly higher training speed, outperforming the CPU-based system by a factor that ranged from 10 to 75 times.*

### 1 Introduction

Recently, computer-aided diagnosis (CAD) has been embedded in the daily clinical routine assisting the detection of breast cancer. Moreover, several studies[1-6] suggest that utilization of CAD systems seems to increase the detection rates of breast cancer[7]. One of the challenges encountered during the design of a CAD system is that, it takes enormous time to optimally train the pattern recognition system which typically lies in its core. Thus, training of a classification scheme on a normal computer may take hours, or even days. However, once training is completed, the characterization of a case takes infinitesimal time.

One of the solutions proposed, so as to tackle the aforementioned problem, is parallel processing typically involving powerful supercomputers, or server clusters. Unfortunately, this kind of hardware is expensive and therefore accessible only to few people. However, a new promising development in this regard is the emergence of consumer-level graphics processing units (GPUs) as a mainstream computing platform[8].

Over the past few years, Graphics Processing Units have evolved from the traditional fixed-function 3D graphics pipelines used as image-synthesis devices, into powerful, programmable, highly parallel computing devices, becoming an increasingly popular tool in many research fields including image analysis. This dramatic shift was the inevitable consequence of consumer demand for videogames, advances in manufacturing technology, and the exploitation of the inherent parallelism in the graphics pipeline[9].

Today, graphics processing units constitute a low-cost, low-power (watts per flop) very high performance alternative to conventional microprocessors. For example, a Geforce 8800 GTX with a theoretical peak 520 GFLOPs (1 GFLOP equals 1 billion floating point operations per second), and dissipating 150 watts, costs about \$200. This is an order of magnitude faster than ordinary CPUs.

Nevertheless, the use of GPUs for general purpose computations in various scientific fields did not begin to gain momentum until the introduction of specialized programming frameworks, such as Stanford University's BrookGPU language[10], NVidia's CUDA (Compute Unified Device Architecture)[11], Microsoft's AP (Accelerator Project)[12], and University of Waterloo's Sh Embedded Meta-programming Language[13], which provided an easy way to harvest the GPU's tremendous parallel computation potential.

Previous studies in the field of image processing and analysis that attempt to benefit in speed from the utilization of GPUs, include implementations of neural networks[14], Kernel methods for Support Vector Machine classifiers[15], k-Nearest Neighbor search methods[16], and algorithms for computed tomography reconstruction[8].

The aim of this study is to propose a GPU-based solution that will accelerate the training of a Probabilistic Neural Network classifier.

## 2 Materials and Methods

The Probabilistic Neural Network (PNN) classifier was introduced by Donald F. Specht back in the late 1980s [17]. Having solid theoretical foundations based on Parzen-Window technique and Bayes theorem, PNN classifier has been used in a wide variety of applications. Equation 1 summarizes the discriminant function of a PNN classifier. Hence for class  $j$ :

$$g_j(x) = \frac{1}{(2\pi)^{p/2} \sigma^p N_j} \sum_{i=1}^{N_j} w(y_i) \quad (1)$$

where  $w(y_i)$  is a function of:

$$y_i = \frac{\sqrt{\|x - x_i\|^2}}{\sigma} \quad (2)$$

In case

$$w(y) = e^{-\frac{y^2}{2}} \quad (3)$$

that leads to a discriminant function with a Gaussian kernel:

$$g_j(x) = \frac{1}{(2\pi)^{p/2} \sigma^p N_j} \sum_{i=1}^{N_j} e^{-\frac{\|x-x_i\|^2}{2\sigma^2}} \quad (4)$$

where  $x$  denotes the test pattern vector to be classified,  $x_i$  the  $i$ -th training pattern vector,  $N_j$  the number of patterns in class  $j$ ,  $\sigma$  a smoothing parameter, and  $p$  is the dimensionality of the feature vector. The test pattern  $x$  is then classified under the class with the larger discriminant function value.

Although, easy training is considered to be one of the most important advantages of the PNN[17], optimal design of a PNN classifier, as a reliable pattern recognition system, typically involves a feature selection and an evaluation techniques. In the case of robust, but computationally intensive techniques, such as the exhaustive search of feature combinations and the leave-one-pattern-out evaluation, are selected the necessity for acceleration of the training procedure becomes apparent.

In order to adapt the training of the aforementioned classifier to the inherently parallel SIMD (Single Instruction Multiple Data) architecture of a GPU, the whole procedure had to be broken into many tasks that run concurrently. In technical terms, the challenge was to design the kernel, or the small code fragment running in multiple threads, in an optimal way in order to evenly distribute the workload and maximize performance.

In the proposed implementation, the first step involved the enumeration of all possible feature combinations. Following the transfer of the training dataset from the memory of the host PC to the GPU's memory, each thread was assigned with a single feature combination. Specifically, the task of each thread, running concurrently, was to train the PNN classifier with this unique feature combination and evaluate its classification accuracy by means of the leave-one-out technique. Upon completion of all threads, results were transferred back to the host's memory for presentation. It should be noted that the NVidia's CUDA GPU programming framework was selected for developing the proposed implementation, mainly due to its maturity.

In order to evaluate its performance, the developed GPU-based classifier was trained and the required training time was measured. The latter was then compared with the respective training time of the same classifier running on a typical CPU and programmed in C programming language. Both the GPU and the CPU-based systems were trained on the same training dataset, comprising actual textural features.

Specifically, a total number of 80 ROIs (40 normal and 40 containing micro-calcifications) were extracted from digital mammograms by an experienced physician. Hence, the training dataset included 80 patterns with each pattern consisting of 20 textural features based on the ROI's image histogram (1st order statistics) and the co-occurrence & run length matrices (2nd order statistics)[18, 19].

All experiments were performed on a desktop PC featuring a Pentium 4 CPU at 3.40GHz with 2GB of RAM, and hosting a GeForce 8800 GT with 512MB of DDR3, which is considered a fairly medium class GPU.

### 3 Results and discussion

Table 1 illustrates the computation time measured by both classifier systems. As far as the GPU is concerned, total processing time includes also the time required for memory transfers between the host's and the GPU's memory. As one can easily observe, the proposed GPU based PNN classifier system achieved significantly lower training times in almost all cases [Figure 1].

Exhaustively Combined Features	Number of Combinations	PNN Training Time (ms)				CPU/GPU Training Time
		CPU	GPU			
			Processing	Transfer	Total	
2 of 20	190	1007.87	97.95	0.79	98.74	10.2
3 of 20	1140	7940.35	274.47	1.33	275.80	28.8
4 of 20	4845	40937.44	1185.85	1.95	1187.80	34.5
5 of 20	15504	153122.98	2979.28	4.15	2983.43	51.3
6 of 20	38760	436336.44	7269.74	7.72	7277.46	60.0
7 of 20	77520	977416.56	12961.83	12.74	12974.57	75.3
Sums	137959	26.9 mins			0.4 mins	65.2

Table 1. Comparison of the computation time required by both systems, given in milliseconds.

The ratio of CPU to GPU training time provides a measure of the achieved acceleration.

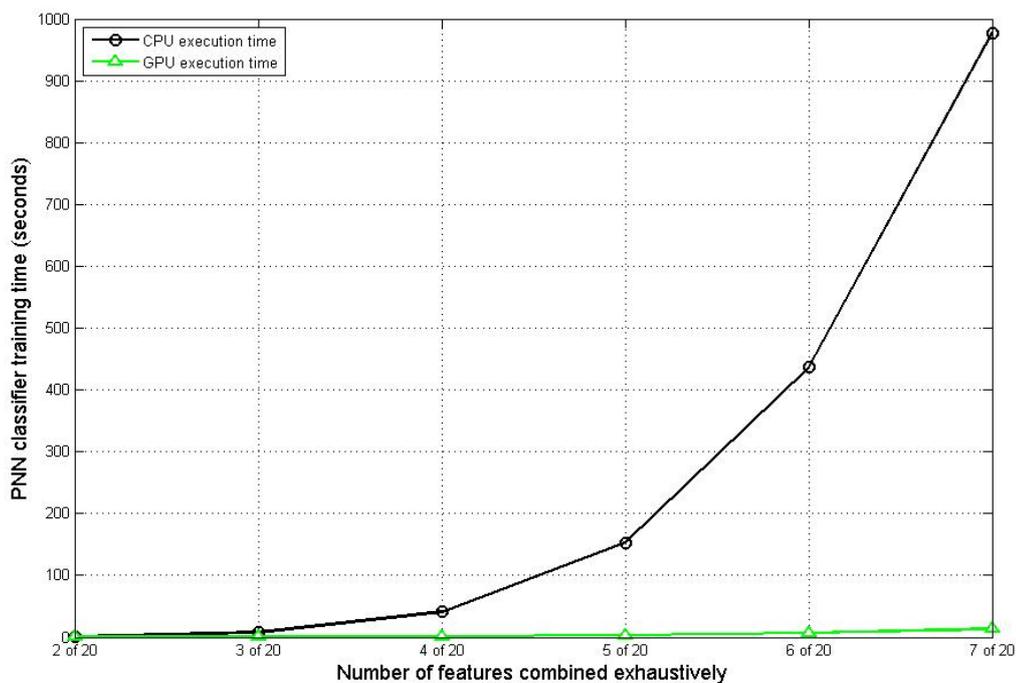


Figure 1. Evolution of the required training time for both the GPU and the CPU based classifiers.

In case of exhaustive combination of 2 out of 20 features GPU performed only 10.2 times faster as depicted in figure 2. This is mainly due to the fact that for 190 feature combinations, only a small number of concurrent threads opened, and the GPU’s processing potential was not fully exploited. On the other hand, best performance enhancement was observed in the case of exhaustive combination of 7 out of 20 features combinations, where the GPU outperformed the CPU 75.3 times by employing 77520 threads.

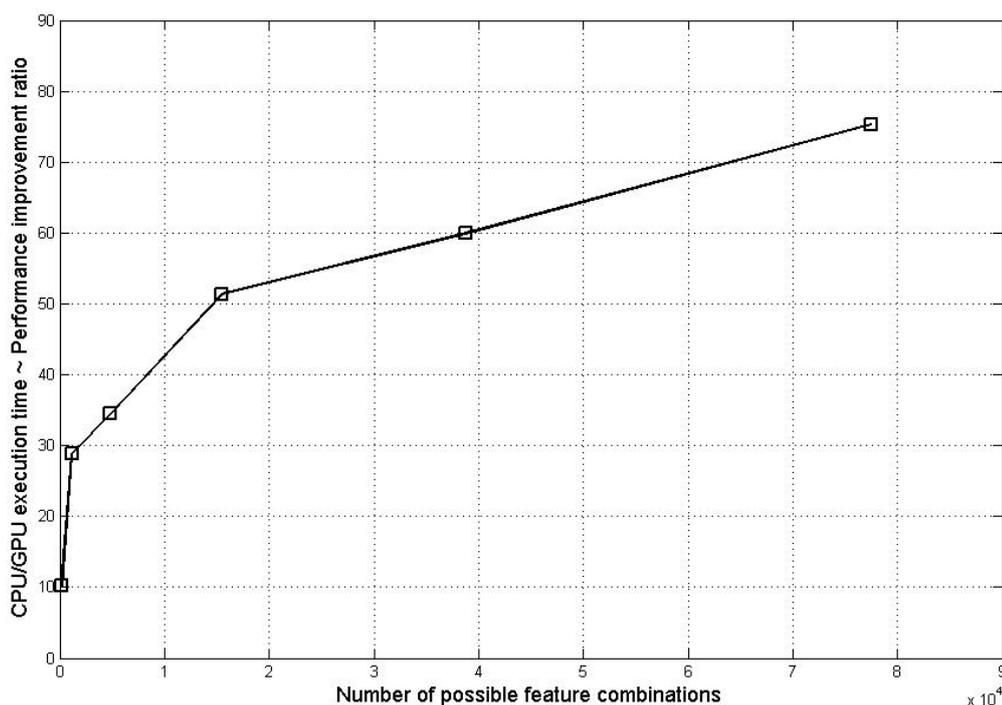


Figure 2. Performance improvement achieved

It should be noted that this speed-up did not have any impact on the classification accuracy of the proposed system, as both systems provided identical results. Because the proposed GPU based classifier system opened one thread per feature combination, it reached the memory limits of the GPU device used. Future memory or algorithm optimization could lead to further increase in performance.

#### 4 Conclusions

According to experimental results, if efficiently programmed, GPUs have the potential to accelerate the training of a PNN classifier. The proposed GPU based classifier system achieved higher training speeds in all cases, peaking its performance as the number of concurrent threads maximized. However there are certain memory issues that, if addressed, can lead to further increase in performance.

#### Acknowledgements

The first author was supported by a grant from the Greek State Scholarships Foundation (IKY).

#### References

- [1] Freer, T. W. and Ulissey, M. J., "Screening mammography with computer-aided detection: prospective study of 12,860 patients in a community breast center," *Radiology*, vol. 220, pp. 781-6, Sep 2001.

- [2] Gur, D., Sumkin, J. H., Rockette, H. E., Ganott, M., Hakim, C., Hardesty, L., Poller, W. R., Shah, R., and Wallace, L., "Changes in breast cancer detection and mammography recall rates after the introduction of a computer-aided detection system," *J Natl Cancer Inst*, vol. 96, pp. 185-90, Feb 4 2004.
- [3] Birdwell, R. L., Bandodkar, P., and Ikeda, D. M., "Computer-aided detection with screening mammography in a university hospital setting," *Radiology*, vol. 236, pp. 451-7, Aug 2005.
- [4] Cupples, T. E., Cunningham, J. E., and Reynolds, J. C., "Impact of computer-aided detection in a regional screening mammography program," *AJR Am J Roentgenol*, vol. 185, pp. 944-50, Oct 2005.
- [5] Dean, J. C. and Ilvento, C. C., "Improved cancer detection using computer-aided detection with diagnostic and screening mammography: prospective study of 104 cancers," *AJR Am J Roentgenol*, vol. 187, pp. 20-8, Jul 2006.
- [6] Morton, M. J., Whaley, D. H., Brandt, K. R., and Amrami, K. K., "Screening mammograms: interpretation with computer-aided detection--prospective evaluation," *Radiology*, vol. 239, pp. 375-83, May 2006.
- [7] Doi, K., "Computer-aided diagnosis in medical imaging: historical review, current status and future potential," *Comput Med Imaging Graph*, vol. 31, pp. 198-211, Jun-Jul 2007.
- [8] Xu, F. and Mueller, K., "Real-time 3D computed tomographic reconstruction using commodity graphics hardware," *Phys Med Biol*, vol. 52, pp. 3405-19, Jun 21 2007.
- [9] Luebke, D. and Humphreys, G., "How GPUs Work," in *Computer*. vol. 40, 2007, pp. 96-100.
- [10] "Stanford University's BrookGPU:  
<http://graphics.stanford.edu/projects/brookgpu/>."
- [11] "NVIDIA's CUDA: <http://developer.nvidia.com/object/cuda.html>."
- [12] "Microsoft Research Accelerator Project: <http://research.microsoft.com/en-us/downloads/648909e1-cb85-46c4-9a94-3cca55971b1d/>."
- [13] "University of Waterloo's Sh Embedded Metaprogramming Language:  
<http://libsh.org/>."
- [14] Kyoung-Su, O. and Keechul, J., "GPU implementation of neural networks," *Pattern Recognition*, vol. 37, pp. 1311-1314, 2004.
- [15] Ohmer, J., Maire, F., and Brown, R., "Implementation of Kernel Methods on the GPU," in *Digital Imaging Computing: Techniques and Applications (DICTA 2005)*, 2005, pp. 543- 550.
- [16] Garcia, V., Debreuve, E., and Barlaud, M., "Fast k nearest neighbor search using GPU," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08*. Anchorage, AK, 2008.
- [17] Specht, D. F., "Probabilistic Neural Networks," *Neural Networks*, vol. 3, pp. 109-118, 1990.
- [18] Haralick, R. M., Shanmugam, K., and Dinstein, I. H., "Textural Features for Image Classification," *IEEE Transactions on Systems Man and Cybernetics* vol. 3, pp. 610-621, 1973.
- [19] Galloway, M. M., "Texture analysis using gray-level run lengths," *Computer Graphics and Image Processing*, vol. 4, pp. 172-179, 1975.