Investigation on Performance and Energy Efficiency of CNN-based Object Detection on Embedded Device

Sangyoon Oh, Minsub Kim, Donghoon Kim
Dept. of Computer Engineering
Ajou University
Suwon, Rep. of Korea
{syoh, skms8622, paian}@ajou.ac.kr

Minjoong Jeong
Dept. of Supercomputing Applications
KISTI
Daejeon, Rep. of Korea
jeong@kisti.re.kr

Minsu Lee
Dept. of Computer Science and Engineering
Ewha Womans University
Seoul, Rep. of Korea
michelle.lee@ewha.ac.kr

***Abstract*—The use of a Convolutional Neural Network based method for object detection increases the accuracy that surpasses human visual system. Because it requires considerable computational capability, its use in embedded devices that place constraints in terms of power consumption as well as computational capability has thus far been limited. However, with the recent development of GPU for use in embedded devices and open-source software library for machine learning, it has become viable to utilize CNN in an energy-efficient embedded computing environment. In this study, CPU and GPU performance and energy efficiency of CNN-based object detection inference on an embedded platform is investigated through comparison with a traditional PC-based platform. Two publicly available hardware platforms are empirically evaluated; in one of them—NVIDIA Jetson TX-1—the results demonstrate image processing performance of 65% of that of the PC, while the embedded device consumes 2.6% of power consumed by the PC.**

Keywords—CNN; embedded device; energy efficiency; object detection; performance

I. INTRODUCTION

Computer vision is the field that studies and develops methods for yielding understanding and insights from digital images or videos. As substantial computational capability is required for processing computer vision algorithms such as those for object detection on images or videos, the application platforms have been limited to stationed machines such as servers and personal computers. Embedded devices that are limited in battery power as well as computational capability (i.e., memory footprint and CPU) have not been considered as computer vision platforms notwithstanding the high demand for the same.

The situation is transforming rapidly, and GPGPU (General-purpose computing on graphics processing units) on embedded devices are now available. By adopting GPGPU, parallel processing methods can be utilized to increase computational capability with even a single CPU-GPU pair. The platform has the potential to yield more computational power through increase in the number of GPUs used.

Another critical research area providing momentum for developing precise object detection applications for images and videos comprises deep learning and deep neural network (DNN). In particular, CNN (convolutional neural network) enables the enhancement of object detection accuracy to the level of recognition by humans. Because of the availability of numerous high-quality open-source software library for deep learning, including Caffe [1], Torch [2], and Tensorflow [3] and their use of GPU, DNN research has progressed extensively.

However, the computing environment of PCs/Servers and embedded computing devices are dissimilar. Thus, the use of these open-source machine learning software libraries in embedded devices is likely to be limited. These open-source libraries are used in PCs/Servers for both training DNN models and running inferences to classify (e.g., object detection). However, contingent on the computational capability that each embedded device exhibits, the use of these open-source libraries for training and building models is not likely to be feasible.

To operate with a deep neural network model (DNN), a neural network is to be trained using labeled training dataset to determine its parameters. Then, the trained network is deployed to run inference to classify or process unknown new inputs. Although GPUs accelerate the speed of the training process remarkably, online DNN training in an embedded environment continues to be challenging owing to high computational cost and substantial energy consumption. Therefore, to utilize a DNN-based model in an embedded environment, the training process is generally performed on a traditional CPU and GPU based platform. Then, the online-inference is carried out in an embedded platform with the trained model.

In this study, CPU and GPU computational performance and energy efficiency of CNN-based object detection inference on an embedded platform are investigated by comparing these parameters with those of a traditional PC-based platform. The

performance and energy efficiency of a DNN-based inference job in CPU-only mode and GPU mode running on two publicly available hardware platforms—an NVIDIA Jetson TX-1 and an NVIDIA GeForce GTX970—were compared. The empirical evaluation results demonstrate that the embedded device consumes 1.5% of the PC power consumption, while it exhibits 65% of the image processing performance of the PC. The results also demonstrate that the use of GPU is highly effective in object detection for both PC and embedded device in terms of performance and energy efficiency.

The remainder of this paper is organized as follows: The experimental design and setup for the investigation is presented in Section 2. The experimental result is presented in Section 3, and Section 4 presents the conclusions.

II. Experiment Design And Setup

A. Experimental Dataset Preparation

In this study, ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 12 dataset [4] was used for the benchmark experiments. ILSVRC12 is one of the representative benchmark datasets for evaluating performance of image classification and localization tasks. The ILSVRC12 dataset was downloaded from [5], and their target label information was downloaded from [6]. The downloaded jpeg format dataset (6.3 GB) includes approximately 50 000 images with varying sizes (Figure 1). To convert the raw image dataset into an input format for the Caffe deep learning framework, which is the open-source software library used for machine learning, an LMDB is created using Python package lmdb and Caffe's python package. The LMDB provides key-value storage, wherein each <key, value> pair is an image in the experimental dataset. Moreover, the image files were resized to 256 × 256 pixels.

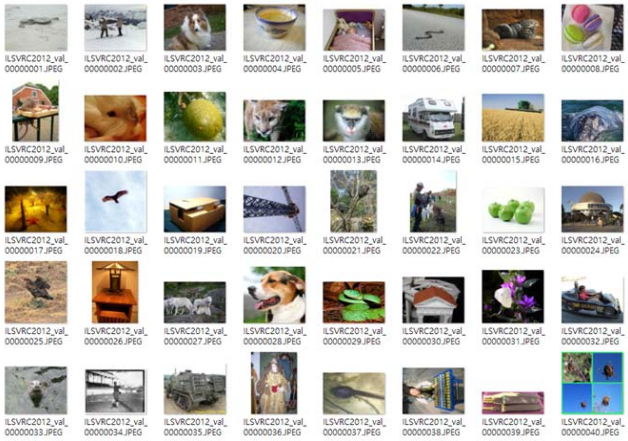

Fig. 1. Example of experimental image datasets

B. Inference Model

Convolutional Neural Networks (CNN) is a special case of neural network that consists of several convolutional layers, subsampling layers, and completely connected layers [7]. The design of a CNN was originally inspired by the visual mechanism in the brain. CNNs have advanced significantly in pattern and image recognition tasks. Recently, object recognition performances of CNN-based methods on the ImageNet dataset have surpassed the human visual system [8].

A trained CNN-based model named GoogLeNet was used for evaluating performance and energy efficiency of object detection inference. GoogLeNet is a 22 layer CNN architecture codenamed "Inception" [9]. The GoogLeNet model achieved the new state-of-the-art for classification and detection in ILSVRC 2014. As the GoogleNet model has a deep and complicated CNN architecture, the model incurs higher computational cost and power consumption. Therefore, the GoogLeNet model is reasonable for comparing conventional and embedded environments in terms of performance and energy efficiency.

We downloaded a replication model [10] for Caffe of the published GoogLeNet[9]. The size of a trained GoogLeNet model with ILSVRC12 dataset is 51 MB; top-1 accuracy of the trained model is 68.4%, and top-5 accuracy on the validation set is 88.4%. Top-*k* accuracy can be computed as the probability of existence of the correct answer among inferred results ranked in top-*k*.

C. Experimental Environment

In this study, to investigate the effectiveness and efficiency of GPUs in embedded environments, the GoogLeNet inference model was run on two hardware platforms—an NVIDIA Tegra TX1 and an NVIDIA GeForce GTX 970. The detailed specifications of the two hardware platforms are described in Table 1. Moreover, to run GPU-enabled Caffe, the CUDA Toolkit 8.0, cuDNN v5.1 library, several dependent libraries, and Caffe were installed separately on two hardware platforms.

To perform a comparative study for evaluating CPU mode and GPU mode in PC and embedded platforms, four experiments were performed that run GoogLeNet inference model with ISLVRC12 dataset in 1) CPU mode on PC platform, 2) CPU + GPU mode on PC platform, 3) CPU mode on embedded platform, and 4) CPU + GPU mode on embedded platform.

TABLE I. Specification of Experimental Hardware Platforms

	Personal Computer	**Embedded Device**
OS	Ubuntu 14.04 LTS	Linux 4 Tegra 24.2.1
GPU	2 x GeForce GTX 970[a] (4GB RAM)	256-core Maxwell GPU (4GB RAM)
CPU	Intel® Core™ i7-5930K @ 3.50 GHz (6Core, 15M)	Quad 64-bit A57 cores + Quad 64-bit A53 cores
RAM	64GB DDR4	4GB LPDDR4
Storage	3 x Seagate 1TB HDD	16 GB eMMC

[a]. Even though it is equipped with two GPU, we have used only one GPU for experiment.

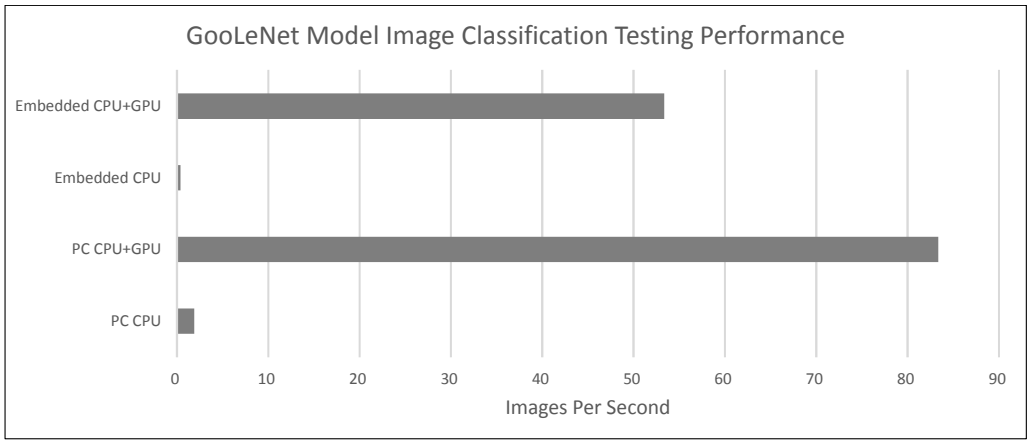

Fig 2. GoogLeNet Model Image Classification Performance Results

D. Evaluation Criteria

To compare the computational performance and energy efficiency of each experiment, we used two evaluation criteria. To examine the execution speed, the number of processed images in a second (images/second) was calculated with the number of total test image files and total execution time for processing test images. Also we measured the power consumption (Watts) with a power meter (HPM-300A) in an idle state and an execution state. Based on these information, energy efficiency (images/second/Watt) was computed.

III. RESULTS

In this study, three performance criteria have been measured through the empirical experiments on a PC and an embedded device. They are accuracy of image classification, computational performance (i.e., number of images processed per second), and energy efficiency. As mentioned, the objectives of these experiments are to measure computational performance and energy efficiency of CNN-based object detection inference on a personal computer and an embedded device (with GPU). Identical GoogLeNet model was used for both the platforms. To evaluate the performance of the deployed GooLeNet models, their accuracy of image classification was evaluated using ILSVRC12 dataset. We confirmed that the accuracies are the same as top-1 68% and top-5 88.4%.

The number of images that can be processed per second was compared among PC with CPU only, PC with CPU + GPU, embedded device (NVIDIA Jetson TX-1 board) with its CPU only, and embedded device with its CPU + GPU. The results are presented in Figure 2 in a bar graph format for convenient comparison. Moreover, the throughput enhancement of each case in Table II was calculated. The comparison demonstrates that the performances of the PC and embedded device in CPU-only mode are in the ratio 4.9:1. This implies that if the image classification job is performed with only CPU, the PC can process approximately five times the number of images that the embedded device processes in an equivalent period. A comparison of the throughput of the CPU + GPU mode with that of the CPU only mode reveals that by utilizing both GPU and CPU, the throughputs of the PC and embedded device are enhanced by factors of 44.4 and 139.4, respectively. This result demonstrates that the performance gain in the embedded device by utilizing GPU is significantly higher than that in the PC. As the throughput of the embedded device with CPU + GPU is increased, the performances of the PC and embedded device in CPU + GPU mode are in the ratio 1.5:1.

TABLE II. IMAGE CLASSIFICATION PERFORMANCE IN IMAGE PER SECOND

HW Platform	**CPU only**	**CPU+GPU**	***Throughput Improvement (PC/Embedded)***
Personal Computer	1.878 (images/sec)	83.351 (images/sec)	44.391 (times)
Jetson TX-1 Embedded Device	0.383 (images/sec)	53.355 (images/sec)	139.354 (times)
Throughput Improvement (PC/Embedded)	4.903 (times)	1.562 (times)	

TABLE III. COMPARISON OF ENERGY CONSUMPTION

	Personal Computer	**Embedded Device**
Power Consumption in **idle state**	1180 (Watts)	10.4 (Watts)
Power Consumption in **full processing.**	1203 (Watts)	27.2 (Watts)
Difference	23 (Watts)	16.8 (Watts)

TABLE IV. COMPARISON OF PROCESSING POWER PER WATT

	Personal Computer	**Embedded Device**
Images per second	83.351 (images/sec)	53.355 (images/sec)
Additional Power Consumption from the GPU use	23 (Watts)	16.8 (Watts)
Processing power per watts (Images per second / additional power consumption)	3.624 (images/sec/Watts)	3.176 (images/sec/Watts)

The next measurement conducted was that of energy efficiency. Although GPUs enables substantial increase of performance of embedded devices, a majority of embedded devices are limited in power consumption. Thus, it is critical to measure the power consumption of each environment. Table III presents the power consumption in idle state and full-processing and their variations.

The processing capabilities of each environment are calculated, and the results are presented in Table IV. As GPU is more energy efficient compared to CPU, adding GPUs to the processing system yields enhanced results in terms of energy efficiency. This implies that GPUs consume less energy or processing an image, and per-watt processing power is higher with GPUs.

IV. Discussion

In this study, performance and energy efficiency of CNN-based object detection inference on both a personal computer and an embedded device (with GPU) have been measured for an objective assessment of the image classification capability of Jetson TX-1 embedded board in terms of performance and energy efficiency. Combined with its Tegra GPU, ARM equipped Jetson TX-1 board produces 65% of the number of images processed per second by the PC with GeForce GTX 970, while its power consumption is 2.6% that of the PC.

The variation between the image processing performances (i.e., images per second) are more extensive in CPU only mode (marginally more than four times) than CPU + GPU (1.5 times). This implies that the performance deficiency of embedded device can be replenished by using GPU. The throughput enhancements of the PC and embedded device with GPU are 44.391 and 139.354 times, respectively. Thus, throughput enhancement with GPU is higher in an embedded device.

Although PC produces enhanced number of images per second performance (i.e., 83.351 vs. 53.355), it consumes between 1180 and 1203 W, whereas the Jetson TX-1 embedded board consumes 27.2W in its full-fledged processing state as well. It is mainly because a PC is a general-purpose machine, and an embedded device is for a specific use. Therefore, the PC consumes more power on its hard disks, optical disks, and other peripherals. Thus, the following facts were deduced: when it is optimized for image classification (e.g., object detection), the embedded device exhibits higher performance to energy consumption ratio than the PC environment.

Investigation on the performance and energy efficiency of Jetson TX-1 was conducted in this study. In future studies, the present authors intend to study the DNN model as well as design an object detection method by integrating a conventional approach with DNN.

Acknowledgment

This research was jointly supported by Basic Science Research Program through the NRF Korea funded by the Ministry of Education (NRF-2015R1D1A1A01059557) and funded by the Korean Government (MSIP) (NRF-2015R1C1A1A01054305). Also, it was supported by the supercomputing application department at KISTI (Korea Institute of Science and Technology Information) (K-17-L01-C03-S01)

References

[1] Caffe: deep learning framework by BAIR, http://caffe.berkeleyvision.org/

[2] Torch: scientific computing framework for LUAJIT, http://torch.ch/

[3] Tensorflow; an open-source software library for machine intelligence, https://www.tensorflow.org/

[4] O. Russakovsky, et al., “ImageNet Large Scale Visual Recognition Challenge”, International Journal of Computer Vision, 115(3):211-252, 2015. doi: 10.1007/s11263-015-0816-y

[5] IMAGENET: Large scale visual recognition challenge 2012 (ILSVRC2012), http://www.image-net.org/challenges/LSVRC/2012/nonpub-downloads

[6] ILSVRC2012 Target label download from caffe.berkeleyvision, http://dl.caffe.berkeleyvision.org/caffe_ilsvrc12.tar.gz

[7] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in Computer Vision–ECCV 2014. Lecture Notes in Computer Science, Springer, vol. 8689, pp. 818–833, 2014,.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026-1034, 2015.

[9] C. Szegedy et al., "Going deeper with convolutions," Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, 2015. doi: 10.1109/CVPR.2015.7298594

[10] BVLC/Caffe, https://github.com/BVLC/caffe/tree/master/models/bvlc_googlenet