Advances in Intelligent Systems Research, volume 172
Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)

Accelerating Convolutional Neural Network Training for Colon Histopathology Images by Customizing Deep Learning Framework

Toto HARYANTO[1], Heru SUHARTANTO[2]*, Aniati MURNI ARYMURTHY[3], Kusmardi KUSMARDI[4]

[1]*Faculty of Computer Science, Universitas Indonesia, Depok, Indonesia*
[2]*Department of Pathology Anatomy Faculty of Medicine, Universitas Indonesia, Jakarta, Indonesia*
**Corresponding author: heru@cs.ui.ac.id*

Abstract
Cancer diagnose based on the histopathology images is still have some challenges. Convolutional Neural Network (CNN) is one of deep learning architecture that has widely used in medical image processing especially for cancer detection. The high resolution of images and complexity of CNN architecture causes cost-intensive in the training process. One way of reducing the training processes time is by introducing parallel processing. Graphics Processing Unit (GPU) is a graphics card which has many processors and has been widely used to speed-up the process. However, the problem in GPU is the limitation of memory size. Therefore, this study proposes alternative ways to utilize the GPU memory in the training of CNN architecture. Theano is one of middle-level framework for deep application. GPU memory is a critical task in training activity and will affect to the number of batch-size. Customizing memory allocation in Theano can be conducted by utilizing library called 'cnmem'. For training CNN architecture, we use NVIDIA GTX-980 that accelerated by customizing CUDA memory allocation from 'cnmem' library located in 'theanorc' file. In the experiment, the parameter of cnmem are chosen between 0 (not apply cnmem) or 1 (apply cnmem). We use image variation from 32x32, 64x64, 128x128, 180x180 and 200x200 pixels. In the training, a number of batch-size is selected experimentally from 10, 20, 50, 100 and 150 images. Our experiments show that enabling cnmem with the value of 1 will increase the speed-up. The 200x200 images show the most significant efficiency of GPU performance when training CNN. Speed-up is measured by comparing training time of GTX-980 with CPU core i7 machine from 16, 8, 4, 2 cores and the single-core. The highest speed-up GTX-980 obtained with enabling cnmem are 4.49, 5.00, 7.58, 11,97 and 16.19 compare to 16, 8, 4, 2 and 1 core processor respectively

Keywords: *cancer, cnmem, histopathology, GPU*

Introduction

Cancer still become one of the diseases with a high rate of mortality in the world. Even, in 2017 there are more than 1,6 million of new cancer cases are found and 600920 cancer death occurred in The United States [1]. The development of digital images has an important role in cancer detection. Since the high resolution of digital images produced by the digital scanner, cancer identification has been conducted automatically via computer-assisted diagnosis (CAD) [2]. Cancer diagnosis via histopathology images with computer-assisted requiring image processing tasks such as image de-noising, pre-processing, feature extraction and machine learning (supervise, unsupervised or semi-supervised learning). The use of histopathology images and automatic detection with computer system still become challenges because of some reasons. High resolution of images, a variation of images, a special pattern of cell and quality of the histopathology will affect classification result [3]. Deep learning as a state of the art in machine learning has widely applied for histopathology image detection, classification, and segmentation. Convolutional Neural Network (CNN) is one of the prominent architecture in medical image analysis, especially for pathology or histopathology images. The ability of CNN for feature extraction with deep layer reveals more information about the images. Even CNN dedicated as the generic feature extraction of many images [4].

Convolution and max-pooling are two tasks in CNN that have a role in generating feature of the images. Besides the advantages provided by CNN, the implementation of CNN for a large scale of the image will take cost-intensive in training process. For instance, convolution multiplication process steps from the real images with a kernel. Basically, the sub from the real images will be multiplied with a kernel (3x3 or 5x5) to produce the convolved feature. Sub-sample images will capture pixel by pixel depend on the kernel will be used. Consequently, if the real image has a large dimension, convolution will spend more time and resources. It the reason why convolution process very sensitive to the size of the images. In CNN, the time

Copyright © 2020 The Authors. Published by Atlantis Press SARL.
This is an open access article distributed under the CC BY-NC 4.0 license -http://creativecommons.org/licenses/by-nc/4.0/.

complexity of the pooling process also will be influenced by the dimension of images. Since feed-forward or one epoch running in CNN, some images are required. A number of images will be taken randomly to update the parameters. The number of images taken is known as batch-size. When training process, batch-size can be determined experimentally according to the number of whole data set and considering the resources are being used. the number of images will impact the memory necessary.

Graphics Processing Unit (GPU) with CUDA programming produced by NVIDIA has been exploited for many research areas on medical image processing. Many tasks such as image segmentation, image registration, image de-noising and image detection running under GPU [5]. GPUs can reduce time consumption in the training process significantly. The trend of GPU usage in deep learning and medical image application had been studied in [6]. However, the size of GPU memory is one the limitation when training with a bigger number of batch-size. Customization of GPU memory will reduce training time and increase efficiency when training. This research will customize the memory usage of GPU NVIDIA GTX-980 when training the CNN model for histopathology image classification. This research also will explore the influence of CPU and GPU GTX-980 in training from speed and accuracy point of view.

The problem in GPU is memory limitation, especially in CNN training. For example, VGGNet trained on 4 GPU machines for handling memory limitation [7]. The feed-forward process requires enough GPU memory. In this research, we custom the value of a memory allocation library to accelerate the training time of our CNN. In this study, we evaluate the performance using speed-up because we want to compare training time between GPU with cnmem customization and our CPUs and to analyze the influence of batch-size and image size to the training time.

RELATED WORKS

In general, deep learning with various of the architecture become state of the art technique in medical image analysis. Even, the convolutional neural network (CNN) is the most popular architecture used [8]. Segmentation is the most task according to the survey meanwhile MRI and pathology are the most popular type of modalities and research field respectively.

The development of deep learning for large-scale image detection or classification has appeared a very significant contribution. According to [9], CNN as one of the deep learning architecture able to reduce error value by five percent for ImageNet classification challenge compared to the traditional classifier. Necrosis detection and cancer classification via the deep convolutional neural network has been done by [10]. In this research, CNN architecture classifies five output of classes. Three classes for cancer classification and two classes for necrosis detection. Overall accuracy for cancer classification and necrosis detection are 0.69 and 0.81 respectively.

Histopathological images of breast tissue are extracted by the convolutional neural network and classified by CNN, support vector machine (SVM) and k-nearest neighbor (KNN) [11]. There are three steps in the training tasks: nuclei detection, pre-training and fine-tuning. Nuclei detection obtained using blob detection method and auto-encoder, applied in pre-training activity. Meanwhile, fine-tuning is the final task to detect and classify histopathology images. According to the research, the network proposed result in better performance compared to another current method.

The Convolutional neural network also applied to computed tomography (CT) images. This images trained on CNN architecture for obtaining segmentation model. In this research, CNN divided into two-step while training process. A total of 2240 images were used for training first CNN and 84000 pixels patch were used for training second CNN resulting final segmentation. The first and second CNN model reveals the accuracy of 95.8% and 96.8% [12]. Another research related to medical images segmentation using the convolutional neural network are [13]–[15].

Cell or nuclei grading has the main role in cancer detection based on medical images. For this purpose, convolutional neural network combine (CNN) with extreme learning machine (ELM) is implemented. For extracting many forms of feature vector with information of contextual, the multiple fully connected are applied in CNN architecture known as (MFC-CNN) [16]. Therefore, the research overall combining MFC-CNN with extreme machine learning (MFC-CNN-ELM) for nuclei grading and result in a good performance.

METHODOLOGY

Dataset

Histopathology datasets are colon histopathology images collected from [15] research with expert annotation. The images comprise two classes of cancer status (benign and malignant). The image's resolution has 520 x 775 pixels dimension, three channels red, green, blue (RGB) with 0.68 µ/mm. The data also as a part of the Gland Segmentation Challenge at MICCAI 2015 contest. The histopathology images used is hematoxylin and eosin (H&E) colouring as seen in Fig. 1.

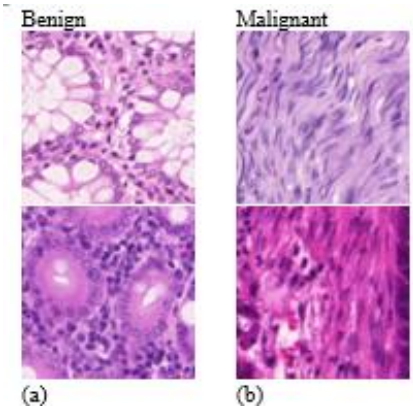


Figure 1. Dataset example of colon histopathology images. (a) benign and (b) malignant

CNN Architecture Description

CNN architecture design in our research comprises of three convolutional layers, three pooling layer, and two fully connected layers. The input layer has various images of the size dimension. We have various of input images size from 32 x 32, 64 x 64, 128 x 128, 180 x 180, and 200x200 images size as input. Our Architecture is described in Fig. 2.

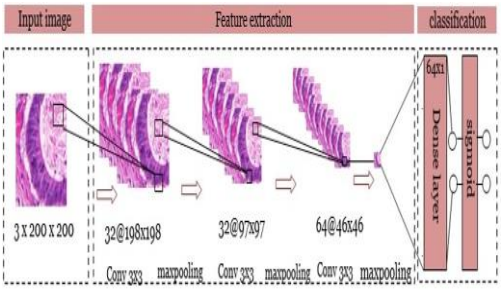


Figure 2. Visualization of proposed CNN architecture

Library cnmem

The cnmem is library for memory management when Deep learning implemented. A Library of cnmem developed with C++ (CPP) programming language and can be customized if we use Keras as our deep learning framework with Theano as the backend of computation. To enable cnmem can modify our .theanorc file and set the value cnmem=[0,1]. The higher value specified will have an impact on a larger percentage of GPU memory allocation.

Hardware and software

This experiment conducted on Intel ® Core i7 ™ 16 cores @3.00 GHz and 65 GB RAM for sequential training. Parallel training running on NVIDIA GPU GTX-980. For deep learning application, Keras API framework with TensorFlow and Theano backend is used. Linux Debian 8 (Jessie) Operating System is used in this work.
GTX-980 as one of the graphics cards designed by NVIDIA. Basically, GTX-980 has 2048 CUDA cores. The basic clock speed of this card is 1126 MHz and can be boosted up to 1216 MHz. Memory specification of this card show in Table 1.

Table 1. Specification of GTX-908

Information	Value
Memory clock	7.0 Gbps
Memory interface	GDDR5
Memory interface width	256-bit
Memory bandwidth	224 (GB/s)
Standard memory config	4GB

Evaluation

To evaluate the performance of our GPU, we use speedup for measurement technique refers to (1).

$$speedup = \frac{t_{(s)}}{t_{(p)}} \quad (1)$$

where $t_{(s)}$ indicate training time on CPU and $t_{(p)}$ is training time via GPU.

RESULT

Performance of GPU GTX-980 without cnmem for various image size

The limitation of GPU for large image datasets is memory usage. In Keras with theano backend, the parameter of cnmem can be tunned to accelerate the training time. This library customized with put the value in a certain range [0,1] or more. The value influences CUDA memory allocation when the training process. The value represents the percent of GPU memory usage. If we choose cnmem = 1, the 95% of GPU memory will be allocated in the training process. Setting cnmem > 1 will affect to the static GPU memory size usage. For instance, cnmem = 4 will spend 4MB GPU memory. Out of memory will occur if the cnmem values are higher than GPU memory available.
Performance of GTX-980 for training CNN architecture on histopathological images was explored with various size of image dimension. Scenarios using mini-batch variations are also considered in training. Overall, the training time without cnmem spends more time as the large as the image dimension increases. Fig. 3 shows the pattern of training time from 32x32 to 200x200 pixel dimension with the number of batch size from 10 to 150 images.

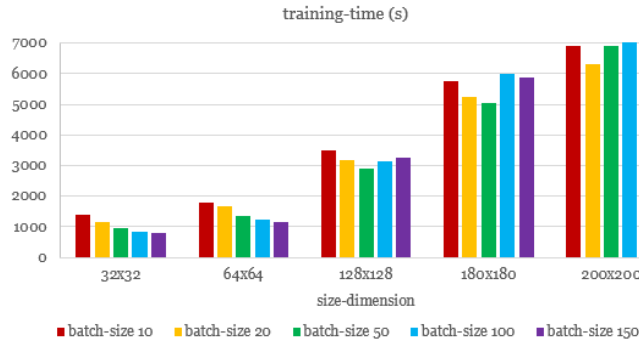


Figure 3. Training using GTX-980 with cnmem = 0 for various images dimension

Performance of GPU GTX-980 with cnmem enable and various image size

Memory efficiency while training CNN is very important because GPU has limited large of memory. enabling the cnmem library with tuning it's parameter experimentally can reduce training time.

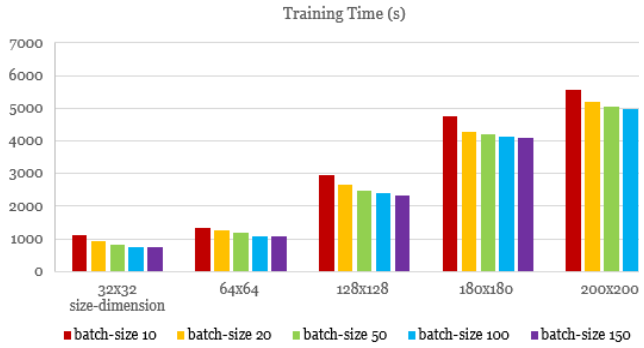
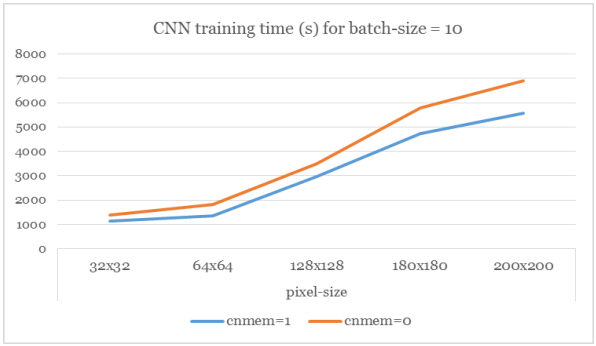


Figure 4. Training using GTX-980 with cnmem = 1 for various images dimension

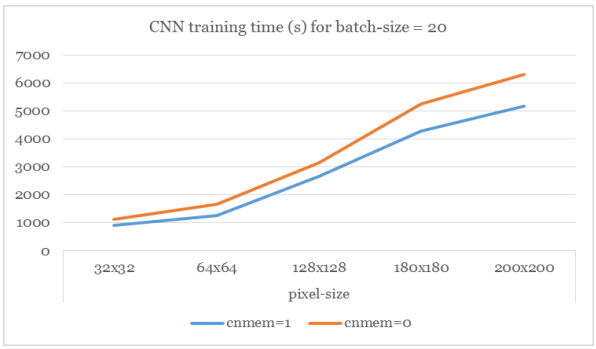
Overall, from Fig. 4 we can see that the training time is reduced when we set the value of cnmem, However, the comparison per image pixel will be discuss in the next sections.

Performance comparison between enabling and disabling cnmem

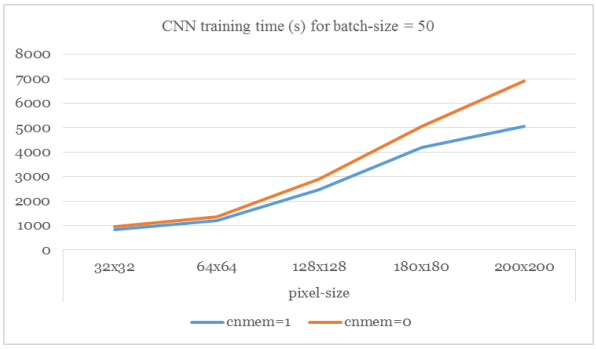
Performance of GTX-980 with and without enabling of cnmem is conducted by alignment the training time between them. For more understanding, visualization in Fig. 5(a)-(e) shows comparison training time in seconds. In these figures, the experiment for training is applied in various batch-size and various image size. The batch-sizes are started from 10, 20, 50,100 and 150.



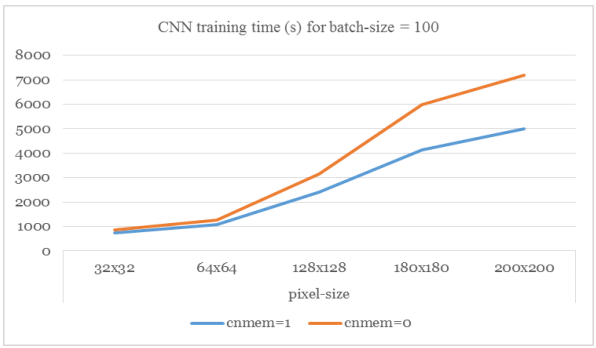
(a)



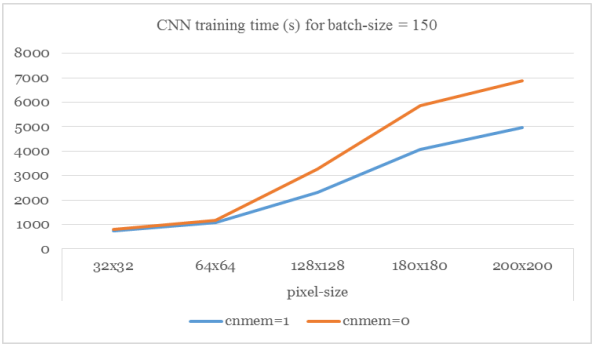
(b)



(c)



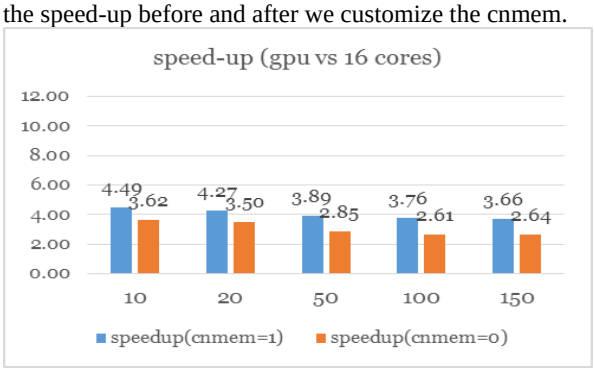
(d)



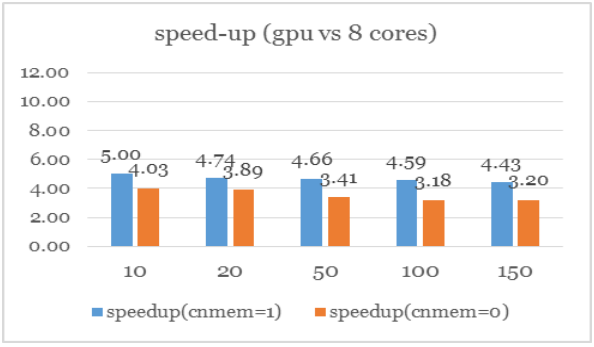
(e)

Figure 5. Comparison between cnmem = 1 (enable) vs cnmem =0 (disable) with various batch-sze from 10(a), 20(b), 50(c), 100(d), and 150(e).

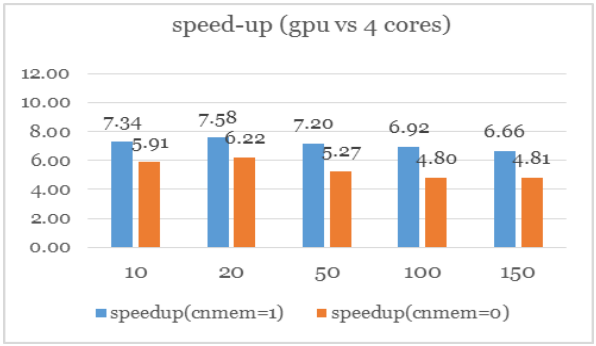
Based on Fig. 5, can be explained that GTX-980 has potential performance if it's memory-optimized by cnmem customization. From the graphic, can be illustrated that the training time not too different significantly on small size image dimension (32 x 32) pixels. The Gap in training time when cnmem are enabled, highly visible on the largest image size (200x200). For this reason, the speed-up analysis for the next experiment focused on this image size. Performance for GPU GTX-980 before and after cnmem applied measured by speed-up. In this case, training via multicores CPU recorded for calculating speed-up. Performance for GPU compared to CPU with various cores from 16, 8, 4, 2 and even single core. Fig. 6(a)-(e) shows the speed-up before and after we customize the cnmem.



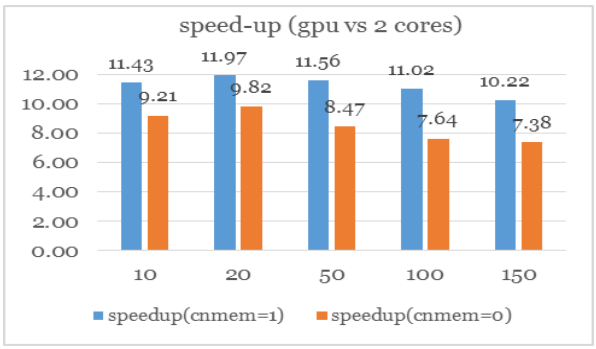
(a)



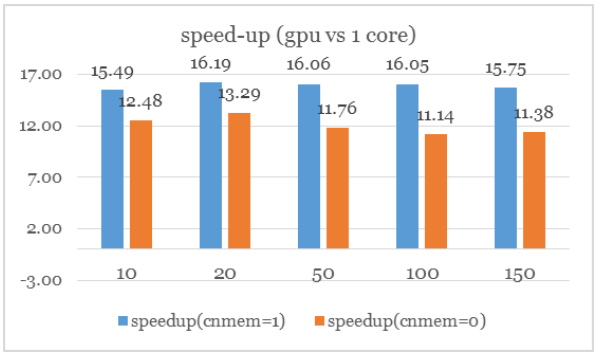
(b)



(c)



(d)



(e)

Figure 6. Speed-up of GPU GTX-980 on histopathological training using CNN compare to CPU from 16 cores (a), 8 cores (b), 4 cores (c), 2 cores (d) and one core (e).

DISCUSSION

In the sequential experiment, Keras framework is implemented on our CNN architecture with TensorFlow backend. CNN has many parameters for being tunned up. On the first experiment scenario, we use Adam as for optimization algorithm.

GPU GTX-908 has 2048 CUDA cores of the processor. However, the size of memory still become the limitation GPU. Therefore, in the next experiment cnmem as the library is customized to explore the performance of GTX-980. According to the previous research in [18], Adam is selected as the optimizer in our training. cnmem basically have to be tunned with the values of it's parameters. By default, the value of cnmem is zero and it is mean that then cnmem is disabled or not used. The value of cnmem is 1

indicates that 95% of memory resources will be utilized for the training.

To ensure the performance of GTX-980 on histopathology image training via CNN, various size of image dimension is measured. The images are captured from smaller to a higher dimension and trained to our CNN architecture proposed. The size of images used on our experiment starts from 32x32, 64x64, 128x128, 180x180 and 200x200 size dimension. In the experiment, we divide into two scenarios. The first scenario, for each image size, is trained and validated with cnmem disable or cnmem=0. Meanwhile, the second one is training CNN via GTX-980 with cnmem = 1 considering the previous experiment.

Fig. 3 and Fig. 4 illustrate the influence of image size dimension to training time without and with enabling cnmem respectively. Overall, The high image size will spend more time to train the CNN models. Images with 200x200 pixel dimension take the slowest training time. From these figures in our experiment, we consider the number of batch-size. Number batch-size represents how many of random image used in feedforward process for updating the parameter.

The influence of batch-size to training time describes in five colors of the graphics. From the figures, we can explain that the use of the small number of batch-size will take more time on the CNN training. These conditions prevail on the majority of the images size dimension with cnmem=1 on Fig 4. However, there is a small anomaly from figure 3 especially on the 128x128, 180x180, and 200x200 images size. In this case, the increasing number of batch-size spend even more time in training exactly. Without a scheduler, the GPU of our server potentially shared with the other user at the same time. This condition caused an anomaly in training time that should be faster with more number of batch-size.

For better understanding, we compare the result from a different viewpoint. Our research is focused on the influence of enabling cnmem on GPU GTX-980. Accordingly, graphics of comparison are necessary. Fig.5 (a)-(e) describes the comparison training time in the trend between cnmem=0 and cnmem=1. The horizontal axis is the image size dimension with five categories of batch-size from 10, 20, 50, 100, 150 and 200. From all the figures show that the highest ‘gap’ occurred on the 200x200 image size. It means that, enabling cnmem give a significant influence on the high dimension of the images. It’s the reason for the last experiment to measure the speed-up only in the 200x200 image size.

The last experiment measures the performance of GPU GTX-980 in CNN training to develop the model and speed-up Speed-up will be calculated based on equation (1). For this purpose, training for sequential computation with any batch-size start from 10, 20, 50, 100 and 150 are conducted on 200x200 histopathology images. Fig. 6 (a)-(e) shows the result performance of GPU GTX-980 machine compare to CPU with various cores.

According to the figure, the customizing of cnmem increase speed-up factor represented by the blue bar in the figures. For CPU computation, by default, all numbers of cores are exploited in the training process. The comparison GPU GTX-980 and CPU conducted with the various number of cores from 16, 8, 4, 2 and single core. The speed-up factor of GTX-980 occurred significantly when compared to a single core of CPU 16.19 times when the value of cnmem = 1 according to Fig. 6(e).

CONCLUSION

The training time of our CNN architecture success to be accelerated by customizing Theano framework on GPU GTX-980 environment. From this study, we can see that the acceleration will impact significantly if we use a higher image dimension. Customizing cnmem library for memory management can increase the speed-up up to 16x when training in GPU compares to the single-core processor of CPU.

ACKNOWLEDGMENTS

This research is supported by Grant of Publikasi International Terindeks 9 (PIT-9) Number NKB-009/UN2.R3.1/HKP.05.00/2019 from Directorate Research and Community Engagement Universitas Indonesia.

REFERENCES

[1] R. L. Siegel, K. D. Miller, and A. Jemal, Cancer Statistics, 2017, Cancer J., vol. 67, 1, (2017), 7–30.

[2] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener, Histopathological Image Analysis: A Review, IEEE Rev. Biomed. Eng., vol. 2, (2009), 147–171.

[3] C. Higgins, Applications and challenges of digital pathology and whole slide imaging, Biotech. Histochem., vol. 90, 5, (2015), 341–347.

[4] L. Hertel, E. Barth, T. Kaster, and T. Martinetz, Deep convolutional neural networks as generic feature extractors, Proc. Int. Jt. Conf. Neural Networks, vol. 2015–Septe, (2015).

[5] A. Eklund, P. Dufort, D. Forsberg, and S. M. Laconte, Medical image processing on the GPU - Past, present and future, Med. Image Anal., vol. 17, 8, (2013), 1073–1094.

[6] T. Haryanto, H. Suhartanto, and X. Lie, Past , Present , and Future Trend of GPU Computing in Deep Learning on Medical Images, in *International Conference on Advanced Computer Science and Information Systems*, 2017, 21–28.

[7] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in *ICLR 2015*, 2015, 1–14.

[8] G. Litjens *et al.*, A survey on deep learning in medical image analysis, Med. Image Anal., vol. 42, December 2012, (2017), 60–88.

[9] L. Bottou, F. E. Curtis, and J. Nocedal, Optimization Methods for Large-Scale Machine Learning, (2018).

[10] H. Sharma, N. Zerbe, I. Klempert, O. Hellwich, and P. Hufnagl, Deep convolutional neural networks for automatic classification of gastric carcinoma using whole slide images in digital histopathology, Comput. Med. Imaging Graph., (2017), 1–12.

[11] Y. Zheng *et al.*, Feature extraction from histopathological images based on nucleus-guided convolutional neural network for breast lesion classification, Pattern Recognit., vol. 71, (2017), 14–25.

[12] Y. Wang, Y. Qiu, T. Thai, K. Moore, H. Liu, and B. Zheng, A two-step Convolutional Neural Network based Computer-aided detection scheme for automatically segmenting adipose tissue volume depicting on CT images, Comput. Methods Programs Biomed., vol. 144, (2017), 97–104.

[13] H. Chen, X. Qi, L. Yu, Q. Dou, J. Qin, and P. Heng, DCAN: Deep contour-aware networks for object instance segmentation from histology images, Med. Image Anal., vol. 36, (2017), 135–146.

[14] J. Xu, X. Luo, G. Wang, H. Gilmore, and A. Madabhushi, A Deep Convolutional Neural Network for segmenting and classifying epithelial and stromal regions in histopathological images, Neurocomputing, vol. 191, (2016), 214–223.

[15] M. Havaei *et al.*, Brain tumor segmentation with Deep Neural Networks, Med. Image Anal., vol. 35, (2017), 18–31.

[16] S. Li, H. Jiang, and W. Pang, Joint multiple fully connected convolutional neural network with extreme learning machine for hepatocellular carcinoma nuclei grading, Comput. Biol. Med., vol. 84, November 2016, (2017), 156–167.

[17] K. Sirinukunwattana, D. R. J. Snead, and N. M. Rajpoot, A Stochastic Polygons Model for Glandular Structures in Colon Histology Images, IEEE Trans. Med. Imaging, vol. 34, 11, (2015), 2366–2378.

[18] D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, in *International Conference on Learning Representations 2015*, 2015, 1–15.