Faster R-CNN Implementation using CUDA Architecture in GeForce GTX 10 Series

Basyir Adam, Fadhlan Hafizhelmi Kamaru Zaman, *Member, IEEE*, Ihsan M. Yassin, *Member, IEEE*, and Husna Zainol Abidin, *Member, IEEE*

***Abstract*—Open-source deep learning tools has been distributed numerously and has gain popularity in the past decades. Training a large dataset in a deep neural network is a process which consumes a large amount of time. Recently, the knowledge of deep learning has been expand with introducing the integration between neural network and the use of graphical processing unit (GPU) which was formerly and commonly known to be used with a central processing unit (CPU). This has been one of the big leap forward in deep learning as it increases the speed of computing from weeks to hours. This paper aims to study the various state-of-the-art GPU in deep learning which included Matrix Laboratory (MATLAB) with Caffe network. The benchmark of the performance is run on three latest series of GPU platforms as of year 2017 by implementing Faster Region-based Convolutional Neural Network (R-CNN) method. Different parameters are varied to analyze the performance of mean average precision (mAP) on these different GPU platforms. The best result obtained in this paper is 60.3% of mAP using the GTX 1080.**

***Index Terms*—Caffe, Convolutional Neural Network, Deep Learning, GPU, MATLAB, Mean Average Precision**

I. Introduction

IN the past few years deep learning has becoming well-known in areas such as computer vision, object localization and image annotation. This is because of its ability to input data is high representation in neural network including Fully connected neural networks (FCNs), AlexNet, Residential network (ResNet) and Long Short Term Memory (LSTM) networks [1]. One of the main contribution in neural network is the GPUs which has reduce the training time of the network significantly as there are many cores and high floating performance in the GPU compared to the CPU. The second contribution is the Compute Unified Device Architecture (CUDA) technology which was proposed by NVIDIA Corporation which open a new opportunity for GPU computing in 2007 and has been developing throughout the years in computing resource [2]. CUDA programming model enables integration between high parallel device kernel C codes with serial host C codes. The third contribution is the implementation of Caffe network and MATLAB interface. Caffe is a framework that is modifiable for deep learning which was written in C++ with MATLAB and Python interface for training convolutional neural networks [3].

The GeForce GTX 10 series that are used for benchmarking are GTX 1060 (3GB), GTX 1070 and GTX 1080. These GPUs are based on Pascal microarchitecture and were released in 2016 which succeeds the GeForce 900 series that are based on Maxwell microarchitecture. The GPUs support CUDA compute capability of version 8.0 as of 2017. The size of video memory of GTX 1060 (3GB) is 3GB whereas GTX 1070 and GTX 1080 have 8GB individually. This is important as larger memory can accumulate more number of datasets and bigger resolution of pictures which outputs a better precision.

Besides that the Faster R-CNN, one of most well-known method which integrates R-CNN with Region Proposal Network (RPN) and hence the name. This method has proved better mAP on object localization which was tested on several datasets including PASCAL Visual Object Classes (VOC) 2007, PASCAL VOC 2012 and Microsoft COCO. The results give mAP of range in 60% to 80% [4].

Having a GPU is very important when training deep neural network as rapid gain in practical is one of the key to build expertise and to solve new problems. This benchmarking results obtained in this paper can serve as a guide for end users to select the best GPU for deep learning.

Throughout this paper, the benchmarking done on these GPUs is by using PASCAL VOC 2007 which consists of about 5000 training images and 5000 testing images with 20 object categories. This paper is arranged as follows: Section I is the introduction of this paper. Section II explains the works related in this paper. After that, Section III explains the implementation of GeForce GTX 10 Series. Section IV shows the results and discussions obtained. Lastly, Section V concludes this paper.

II. Related Works

A. Caffe

Caffe framework was designed to provide a complete set of toolkits with deployed models [3]. These are numbers of highlights of the importance of Caffe:

1. Test Coverage – Caffe will test all new codes in a single

This paper was submitted for review on November 2017. Accepted for publication on 10th April 2018. This research work was financially supported by Research Acculturation Grant Scheme (600-RMI/RAGS 5/3 (16/2015)).

Basyir Adam, Fadhlan Hafizhelmi Kamaru Zaman, Ihsan M. Yassin, Husna Zainol Abidin are with Faculty of Electrical Engineering, Universiti Teknologi MARA, Malaysia (e-mail: basyir_kael@yahoo.com, fadhlan@salam.uitm.edu.my,ihsan_yassin@salam.uitm.edu.my, husnaza@salam.uitm.edu.my).

module to be accepted into a project.

2. Pre-trained models – ImageNet and R-CNN reference models are provided by Caffe with variations.
3. MATLAB and Python bindings – Caffe networks can be constructed in these languages for classifying inputs. Caffe is better known to be constructed using Python language.
4. Modularity – Caffe allows extension to be easy for new network layers, data formats and loss functions.
5. Separation of implementation and representation – Caffe is written as configuration files by using the Protocol Buffer programming language.

Fig. 1 below shows one of the application that uses Caffe framework which is the R-CNN which was later updated to Faster R-CNN. R-CNN combines bottom-up region-based proposals with features computed by CNN. Initially, R-CNN computes the extracted proposals from the input image using selective search technique and pass the proposals to CNN for classification.

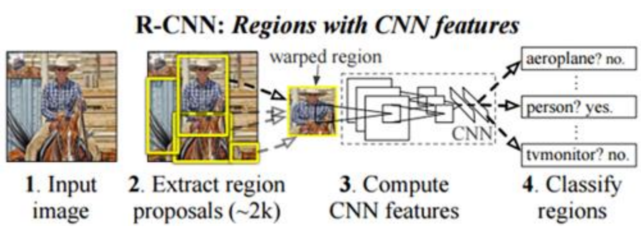

Fig. 1. Flow of R-CNN System that implements Caffe [3, 5]

B. CUDA Technology

CNN is one of the most intensive computation task as it demands a large amount of memory storage and computational power. CUDA is implemented to optimize usage of heterogeneous computation resources (CPU-GPU) and reduce time for classification. Fig. 2 shows matrix multiplication based convolution (ConvMM) with the implementation of CUDA with MATLAB interface and other frameworks to import trained parameters for identical classifier.

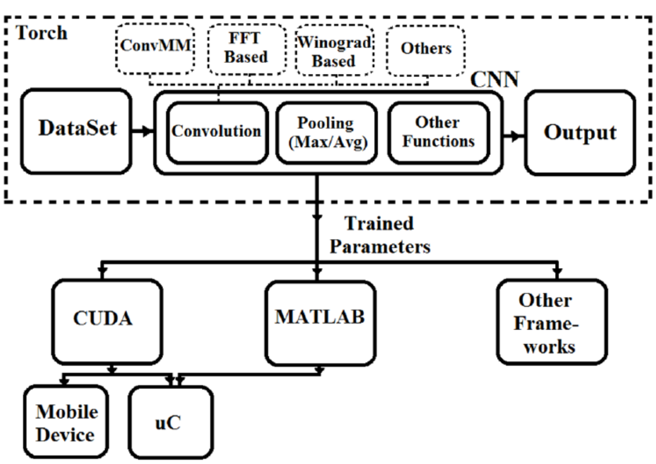

Fig. 2. Block Diagram of a Simple Trained Network involving CUDA [6]

CUDA helps optimizing ConvMM layer by parallelizing transformation step and eliminate excessive data transfers on GPU. The excessive data in the memory is removed due to avoid over-allocated as this can reduce the amount of Random Access Memory (RAM) which lead to reduction of the performance of the computation.

C. Faster R-CNN

The development of Faster R-CNN begins Deep ConvNet, updated to R-CNN, Fast R-CNN and the current version as of 2017, Faster R-CNN which was initially developed by Ross Girshick [4]. This method combines Caffe together with SelectiveSearch and EdgeBoxes techniques to perform object localization and recognition effectively [7, 8]. Fig. 3 displays the overview of Faster R-CNN which implements RPN to form a unified network for detecting objects. This integration between the two networks succeeded the previous version as RPN helps Fast R-CNN by proposing a region so that the system can detect objects easier. Moreover, RPN shares the convolutional computations which reduces the cost of fully connected layers.

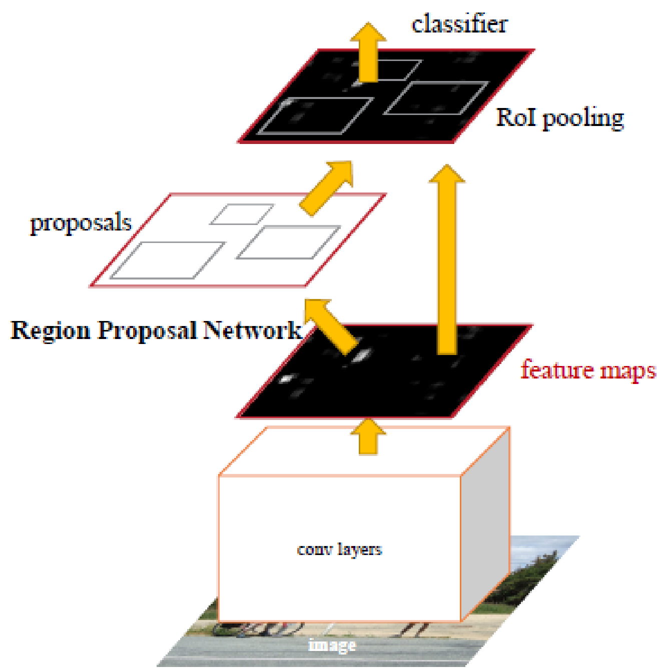

Fig. 3. Faster R-CNN network with RPN module [4]

D. PASCAL VOC 2007 Datasets

PASCAL VOC is a benchmarking for visual object detection and recognition by providing the standard dataset of images with its annotations [9]. Zeiler and Fergus (ZF) network which has three fully connected layers and five convolutional layers is used to help train the network and primarily evaluate the mAP for object detection [10].

Below is the categories used to recognize multiple objects in a single image from the datasets:

1. Airplane.	2. Bike.	3. Bird.	4. Boat.
5. Bottle.	6. Bus.	7. Car.	8. Cat.
9. Chair.	10. Cow.	11. Dog.	12. Horse.
13. Mobile.	14. Person.	15. Plant.	16. Sheep.
17. Sofa.	18. Table.	19. TV.	20. Train.

III. IMPLEMENTATION OF GEFORCE GTX 10 SERIES

This paper introduces the benchmarking of the current GPUs to Faster R-CNN with the current CUDA version 8.0. GPU platform gain a better efficiency than CPU even with a 16 cores of CPU [11]. This is because GPU memory has a better computing efficiency as a GPU may has over a thousands of cores while the current CPU has a maximum of 16 cores as shown in Fig. 4. CUDA enables a pool of shared memory between the GPU and the CPU which can be accessed by both memories [12]. Therefore, data synchronization between both memories after kernel execution is important.

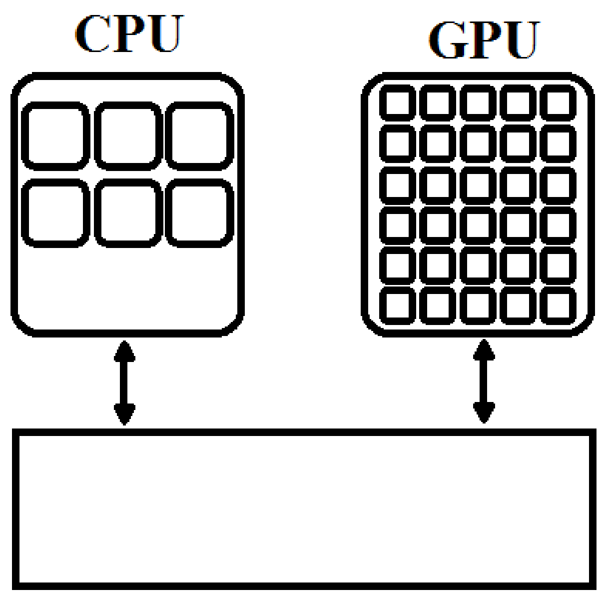

Fig. 4 An illustration of Unified Memory [6]

Fig. 5 shows an example of a graphics card which is GTX 1080, one of the top tier graphics card in the market and highest computational power compared to GTX 1070 shown in Fig. 6 and GTX 1060 (3GB) shown in Fig. 7.

Fig. 5. An example of GTX 1080 Graphics Card [13]

Fig. 6. An example of GTX 1070 Graphics Card [14]

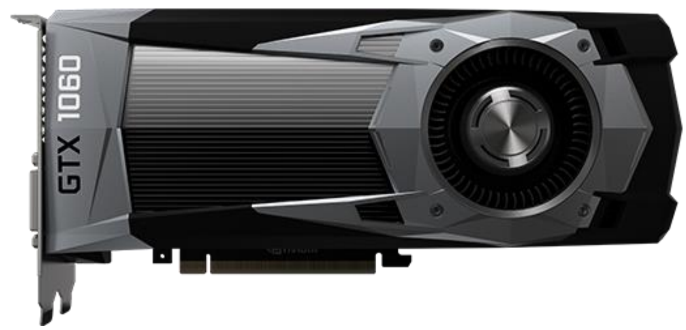

Fig. 7. An example of GTX 1060 Graphics Card [15]

Table I shows the comparisons between the three GPUs of Pascal architecture. GTX 1080 has the overall highest specifications followed by GTX 1070 and GTX 1060 (3GB). In addition, GTX 1080 performs the best in most CNN due to its overall higher throughput and rate of convergence. However, it requires more power consumption and the cost is expensive compared to the other two GPUs.

PASCAL VOC 2007 consist of 10000 images, 5000 used for training images and another 5000 used for testing images. The mAP obtained is the proportion of all 20 categories above the rank which are positive [9].

TABLE I
SPECIFICATIONS OF GTX 1060 (3GB), GTX 1070 AND GTX 1080

GPU (GTX)	1060 (3GB)	1070	1080
Streaming Multiprocessors	9	15	20
CUDA Cores	1152	1920	2560
Base Clock (MHz)	1506	1506	1607
GPU Boost Clock (MHz)	1708	1683	1733
Processing Power (GFLOPs)	3935	6463	8873
Texture Units	72	120	160
Texel fill-rate (Gigatexels/sec)	108.4	180.7	277.3
Memory Clock (MHz)	8000	8000	10000
Memory Bandwidth (GB/sec)	192	256	320
Render Output Unit	48	64	64
L2 Cache Size (KB)	1536	2048	2048
Thermal Design Power (Watts)	120	150	180
Transistors (billion)	4.4	7.2	7.2
Die Size (mm^2)	200	314	314
Manufacturing Process (nm)	16	16	16

IV. Results and Discussions

The results of benchmarking between the GPUs is experimented by observing the mAP. The mAP is influenced by tweaking three parameters in the system. The three parameters are the minimum batch size of the samples, the maximum pixel size of input images and the image scales (short edge of input images).

Table II shows that the higher number of minimum batch size slightly increases the mAP. Since the GTX 1080 and GTX 1070 have 8GB video random access memory (VRAM), the maximum pixel size set for both GPUs was 1000 pixels whereas the GTX 1060 (3GB) has 3GB VRAM, hence the maximum pixel size need to be reduced to 800 pixels but the image scales of all GPUs are set to 600 pixels. The highest mAP obtained by GTX 1080 is 59.7%, GTX 1070 is 59.8% and GTX 1060 (3GB) is 60.2%. The comparison in Fig. 8 shows that the mAP between the three GPUs are not very apart in this parameter.

On the other hand, Table III shows the results of maximum pixel size on the GPUs. The higher resolution of this parameter also slightly increases the mAP. The GTX 1080 and GTX 1070 both can accumulate of 950 pixels of images and more as recorded while GTX 1060 (3GB) is 750 pixels which will decrease its performance of the mAP but all the GPUs support a minimum batch size of 128 and image scales of 600 pixels. The highest mAP obtained by GTX 1080 is 60.3%, GTX 1070 is 59.8% and GTX 1060 (3GB) is 59.3%. GTX 1080 performs a slight better than GTX 1070 and GTX 1060 (3GB). The comparison in Fig. 9 shows that the GTX 1060 (3GB) does not support images of the datasets more than 750 pixels size while GTX 1080 and GTX 1070 could accumulate more pixels in size.

Lastly, Table IV shows the results on the image scales. Image scales is one of the most important parameter as increasing it moderately increases the mAP of the system. In this experiment, all the GPUs support the maximum pixel size of 1000 pixels and minimum batch size of 128. The highest mAP obtained by GTX 1080 is 59.2%, GTX 1070 is 59.4% and GTX 1060 (3GB) is 58.1%. The GTX 1080 and GTX 1070 support a scaling of images at 600 pixels as recorded in the first experiment but the GTX 1060 (3GB) VRAM supports maximum image scales of 500 pixels and below as shown in Fig. 10 which decreases its performance compared to the other two GPUs.

These results obtained by the GPUs are influenced by the computational power of the GPUs. In the aspect of memory clock speed, GTX 1080 is 10000MHz which is slightly faster than the other two GPUs which is 8000MHz. Besides that, the memory bandwidth of GTX 1080 is 320GB/sec, GTX 1070 is 256GB/sec and GTX 1060 (3GB) is 192GB/sec. GTX 1080 has the fastest memory bandwidth.

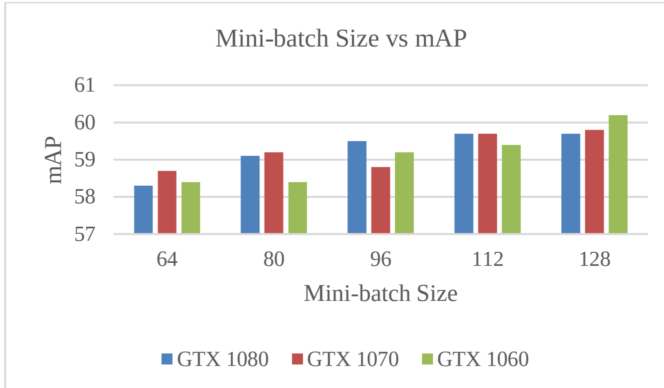

Fig. 8. Performance comparison of mini-batch size on GPU platforms

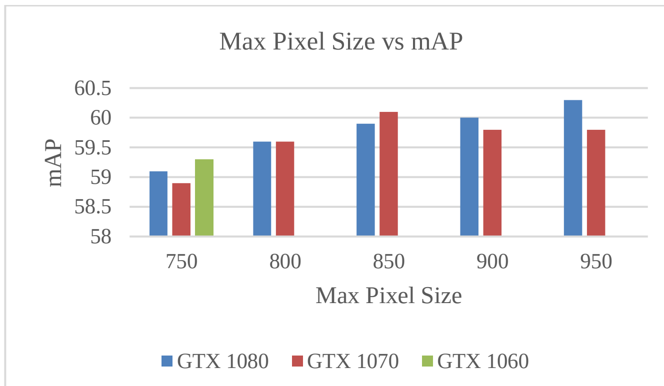

Fig. 9. Performance comparison of maximum pixel size on GPU platforms

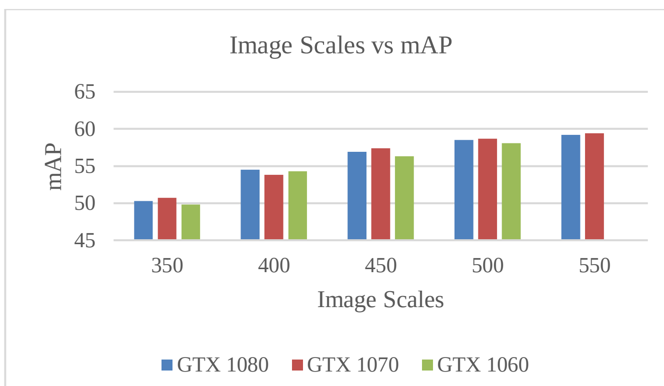

Fig. 10. Performance comparison of image scales on GPU platforms

TABLE III
Results on Benchmarking of Minimum Batch Size

Category	Mini-batch Size														
	GTX 1080					GTX 1070					GTX 1060 (3GB)				
	64	80	96	112	128	64	80	96	112	128	64	80	96	112	128
mAP	58.3	59.1	59.5	59.7	59.7	58.7	59.2	58.8	59.7	59.8	58.4	58.4	59.2	59.4	60.2
Aero	65.3	63.3	65.8	65.1	65.7	62.5	65.5	63.7	64.5	64.4	63.1	63.8	63.6	63.3	65.2
Bike	71.3	71.7	69.5	73.6	70.5	70.2	68.6	69.9	69.3	70.6	69.8	68.8	70.2	72.3	74.3

Bird	56.5	55.4	56.9	53.9	56.4	55.3	54.6	54.8	54.7	57.6	55.9	54.7	57.7	55.8	58.2
Boat	44.3	43.0	44.3	42.0	43.4	40.9	44.4	42.9	44.4	45.4	44.4	43.6	45.2	45.5	44.8
Bottle	28.0	30.0	30.8	31.5	62.3	31.2	29.7	30.7	32.1	31.7	28.3	30.3	33.9	31.1	33.4
Bus	66.7	66.8	69.0	66.8	69.2	65.5	65.9	66.6	68.1	65.3	66.7	64.0	66.0	65.5	67.3
Car	72.3	72.9	73.7	73.5	73.2	73.1	73.4	73.6	73.4	74.0	72.5	72.6	73.6	72.9	73.3
Cat	71.5	71.0	71.6	72.9	71.9	73.2	69.9	71.3	72.8	72.3	70.8	69.2	69.9	72.3	71.5
Chair	33.1	33.3	35.6	35.7	34.7	34.8	35.4	34.7	36.4	36.2	35.3	36.1	34.2	38.2	38.4
Cow	63.3	65.0	64.7	65.6	64.3	66.3	67.8	62.3	66.2	65.6	62.7	67.9	64.6	66.2	66.1
Table	58.7	62.1	62.6	64.4	62.4	63.4	63.1	61.8	61.4	61.4	62.2	59.5	61.9	62.6	62.2
Dog	63.6	67.6	67.6	68.4	67.5	64.9	65.1	65.6	66.9	69.9	66.1	66.0	67.6	65.0	68.2
Horse	75.0	76.7	76.0	79.0	75.5	75.3	78.8	77.1	76.8	78.6	74.5	76.5	76.5	75.7	77.9
Motorbike	68.4	67.5	67.8	66.3	66.9	67.7	69.6	68.1	69.8	69.7	66.9	67.8	67.9	68.6	66.9
Person	64.5	64.3	64.7	64.9	65.3	64.0	64.9	64.1	64.5	65.8	63.6	64.4	64.7	64.3	64.9
Plant	28.2	30.4	26.6	30.8	28.7	29.5	29.2	30.2	29.3	31.1	27.8	27.3	27.5	29.6	29.2
Sheep	56.8	59.3	60.4	58.7	59.9	56.4	61.7	57.0	59.4	58.5	57.5	57.1	57.1	60.4	58.2
Sofa	54.0	55.5	56.4	52.6	56.6	54.2	54.1	54.9	56.9	54.0	55.0	54.1	54.4	50.4	55.2
Train	69.0	68.8	68.8	69.9	71.2	72.0	68.6	71.2	69.7	69.1	70.2	68.7	70.0	71.2	69.9
TV	55.8	57.7	57.4	59.2	59.2	53.5	54.0	56.4	57.4	55.4	54.3	56.2	58.0	55.7	58.4

TABLE IIII
RESULTS ON BENCHMARKING OF MAXIMUM PIXEL SIZE

Category	Maximum Pixel Size														
	GTX 1080					GTX 1070					GTX 1060 (3GB)				
	750	800	850	900	950	750	800	850	900	950	550	600	650	700	750
mAP	59.1	59.6	59.9	60.0	60.3	58.9	59.6	60.1	59.8	59.8	54.2	56.2	57.1	58.4	59.3
Aero	62.6	64.6	63.4	64.9	66.6	64.4	64.8	63.7	65.4	64.4	56.1	59.1	61.9	63.6	62.7
Bike	68.7	70.5	73.4	69.8	71.7	70.7	72.3	73.1	73.9	70.4	66.1	69.7	68.9	71.6	71.6
Bird	56.6	56.7	58.3	58.1	55.8	58.3	56.0	57.8	55.3	57.5	50.3	52.3	55.3	57.6	57.6
Boat	39.5	47.5	42.6	44.0	46.0	43.3	40.5	41.4	45.6	42.5	40.6	40.2	38.7	43.8	44.6
Bottle	32.4	30.0	31.1	31.4	32.4	29.8	31.1	31.3	33.4	30.9	26.5	26.9	29.4	27.9	29.4
Bus	65.6	64.8	67.6	64.2	66.6	62.9	66.7	66.8	65.5	64.3	59.3	65.3	63.5	65.6	63.1
Car	74.0	73.5	73.7	73.6	73.7	73.3	72.5	73.5	74.1	74.4	70.0	72.2	71.9	73.4	74.0
Cat	71.2	70.1	73.4	72.7	74.8	70.8	72.2	71.5	70.9	72.9	60.7	65.1	67.7	70.3	69.0
Chair	35.7	36.8	35.9	35.9	36.4	34.4	35.1	36.6	36.0	38.0	33.2	32.8	32.9	34.0	35.5
Cow	64.6	66.0	67.0	66.4	65.9	65.1	67.2	67.2	64.9	64.0	59.7	62.4	63.8	65.0	66.0
Table	62.1	61.6	64.3	61.4	62.0	59.5	61.6	62.8	61.8	62.5	58.3	58.8	56.7	61.6	60.7
Dog	67.8	68.2	67.6	67.8	68.5	67.7	69.1	69.5	67.5	67.7	60.1	61.3	65.6	66.2	68.5
Horse	76.4	78.4	76.6	77.5	78.1	74.8	77.2	77.5	77.5	78.8	73.1	74.0	75.6	75.8	76.5
Motorbike	69.7	68.5	70.7	69.5	70.5	67.7	68.8	68.3	68.3	67.0	60.2	64.6	66.5	69.1	68.6
Person	65.0	64.6	64.5	64.6	64.8	64.5	64.5	65.0	64.9	65.1	62.3	63.1	64.8	63.9	64.7
Plant	29.9	29.8	31.2	30.5	31.0	29.2	30.5	32.9	29.6	30.8	25.8	29.6	28.3	30.3	29.9
Sheep	57.6	58.4	58.6	59.3	58.6	57.5	57.4	59.5	57.6	56.9	55.7	58.4	56.7	58.3	58.7
Sofa	54.2	55.3	52.9	58.8	56.1	55.2	55.1	57.5	55.0	55.4	47.1	48.8	50.8	47.4	56.5
Train	71.2	69.3	69.5	71.8	69.9	73.1	70.9	70.4	70.3	71.8	63.1	67.8	67.7	69.2	71.3
TV	58.0	56.4	56.3	58.4	57.3	55.1	57.8	55.5	58.0	60.3	55.2	51.7	54.8	53.9	58.0

TABLE IV
RESULTS ON BENCHMARKING OF IMAGE SCALES

Category	Image Scales														
	GTX 1080					GTX 1070					GTX 1060 (3GB)				
	350	400	450	500	550	350	400	450	500	550	300	350	400	450	500
mAP	50.3	54.5	56.9	58.5	59.2	50.7	53.8	57.4	58.7	59.4	45.9	49.8	54.3	56.3	58.1
Aero	52.2	58.7	60.0	63.3	62.6	56.2	55.2	60.5	63.9	64.2	50.4	53.6	56.0	61.9	65.3
Bike	61.9	65.9	68.5	71.8	71.6	62.8	66.2	70.0	71.0	72.8	53.6	59.3	66.0	69.7	70.6
Bird	47.6	51.5	55.1	53.7	55.4	44.8	48.6	54.6	56.1	56.3	38.7	45.6	53.5	55.4	52.6
Boat	33.3	40.4	40.8	42.8	43.0	34.4	37.0	39.7	44.1	44.1	31.2	31.3	37.8	38.1	41.3
Bottle	25.9	27.5	28.9	29.3	29.1	26.0	26.2	29.2	30.0	29.3	24.3	25.7	28.1	28.3	31.1
Bus	54.2	60.1	63.1	63.3	65.3	57.8	59.7	65.7	62.1	64.9	48.9	55.4	58.3	61.8	63.4
Car	68.0	70.7	71.8	73.6	73.7	67.2	71.6	71.6	73.5	73.9	62.9	67.7	70.7	72.2	72.7
Cat	54.9	62.3	65.3	71.5	71.1	54.7	63.2	68.0	69.1	70.0	48.0	54.1	61.0	67.5	68.4
Chair	29.5	31.4	33.9	34.3	35.0	27.9	31.5	34.8	34.3	37.0	27.6	28.2	32.2	33.3	34.0
Cow	57.2	63.5	64.8	66.7	65.0	56.3	63.4	66.1	65.4	65.0	53.2	57.6	63.7	60.4	63.3
Table	51.8	56.0	59.9	59.8	62.6	51.5	54.6	60.6	60.2	62.2	43.0	54.1	57.7	58.3	63.7
Dog	50.6	55.5	62.1	64.2	66.2	49.9	59.1	62.2	68.0	66.3	43.5	44.3	57.6	62.5	64.9
Horse	65.7	74.2	75.4	75.3	76.6	69.5	72.7	73.7	75.5	77.9	66.0	66.1	71.6	75.0	75.5
Motorbike	59.1	61.1	66.8	66.4	69.3	58.7	62.5	65.9	68.2	67.0	55.3	61.7	63.7	66.5	66.8
Person	59.3	62.9	63.7	64.5	64.5	59.4	61.6	63.4	64.4	64.5	56.1	59.6	61.9	63.5	64.1
Plant	25.2	27.6	28.5	29.7	33.3	24.6	27.7	28.7	30.1	31.7	22.2	23.9	24.3	27.0	28.2
Sheep	55.7	53.6	54.7	59.7	56.9	53.2	53.8	59.5	59.1	60.0	51.4	54.4	56.7	55.6	57.3
Sofa	40.4	46.8	50.1	52.6	58.0	45.4	44.3	50.4	53.0	54.2	38.7	42.0	46.6	50.2	53.2
Train	61.4	67.1	68.7	70.0	70.3	63.3	64.2	68.8	68.7	69.7	55.1	62.1	65.7	67.3	68.0
TV	51.1	54.0	55.5	57.7	55.1	50.5	52.3	54.0	57.9	56.4	47.3	49.5	52.8	52.1	58.1

TABLE V
RECOMMENDED GPU FOR DEEP LEARNING

	GTX 1080	GTX 1070	GTX 1060 (3GB)
Cost	Expensive	Medium	Cheap
Efficiency	High	Medium	Low
Big Datasets	Large	Medium	Small
Overall Best (By small margin)	Recommended for Deep Learning but expensive in price	Another recommended for Deep Learning and cost efficient	Not recommended since it does not work with big datasets

V. CONCLUSION

In conclusion, the presented benchmarking performance of these GPUs with the three parameters could achieve the highest mAP of 60% in average on PASCAL VOC 2007 dataset. Using a higher performance GPU such as the GTX 1080 could achieve a better mAP in average if the best parameters are set up due to its specifications. This is because one of the most important factor that is required in a GPU is its memory size to accumulate many sample of datasets with high resolution of images and memory bandwidth to speed up the performance. In spite of that, the cost for a higher performance GPU is very expensive so it is best to find an efficient but cheap GPU in the market. Therefore, the recommended GPUs for Deep Learning are GTX 1080 and GTX 1070 as shown in Table V.

ACKNOWLEDGMENT

I would like to acknowledge PASCAL VOC 2007 for providing the datasets for open-source use. I also would like to thank Faculty of Electrical Engineering, UiTM for management support throughout this project.

REFERENCES

[1] S. Shi, Q. Wang, P. Xu, and X. Chu, "Benchmarking State-of-the-Art Deep Learning Software Tools," in *Proc. 2016 7th International Conference on Cloud Computing and Big Data (CCBD)*, Macau, 2016, pp. 99-104.

[2] Y.-b. Huang, K. Li, G. Wang, M. Cao, P. Li, and Y.-j. Zhang. (2015). Recognition of convolutional neural network based on CUDA Technology. [Online]. Available: https://arxiv.org/abs/1506.00074

[3] Y. Jia *et al.*, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proc. of the 22nd ACM international conference on Multimedia*, 2014, pp. 675-678.

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Proc. Advances in Neural Information Processing Systems 28*, 2015.

[5] R. B. Girshick, "Fast R-CNN," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440-1448.

[6] S. Rizvi, G. Cabodi, and G. Francini (2017). Optimized Deep Neural Networks for Real-Time Object Classification on Embedded GPUs. *Applied Sciences*. [Online]. 7(8), pp. 826. Available: http://www.mdpi.com/2076-3417/7/8/826/pdf

[7] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. (2013). Selective Search for Object Recognition. *International Journal of Computer Vision.*. [Online]. 104(2), pp. 154-171. Available: www.huppelen.nl/publications/selectiveSearchDraft.pdf

[8] C. L. Zitnick and P. Dollár, "Edge Boxes: Locating Object Proposals from Edges," in *Proc. European Conference on Computer Vision*, 2014, pp. 391-405.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2010). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*. [Online]. 88(2), pp. 303-338. Available: host.robots.ox.ac.uk/pascal/VOC/pubs/everingham10.pdf

[10] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," *in Proc. European conference on computer vision*, 2014, pp. 818-833.

[11] N. Vasilache, J. Johnson, M. Mathieu, S. Chintala, S. Piantino, and Y. LeCun. (2014). Fast Convolutional Nets With fbfft: A GPU Performance Evaluation. [Online]. Available: https://arxiv.org/pdf/1412.7580

[12] A. A. Huqqani, E. Schikuta, S. Ye, and P. Chen. (2013). Multicore and GPU Parallelization of Neural Networks for Face Recognition. *Procedia Computer Science*. [Online]. 18, pp. 349 - 358. Available: https://www.sciencedirect.com/science/article/pii/S1877050913003414/pdf?md5=067af0695ea8598c8dff2d6f676ef350&pid=1-s2.0-S1877050913003414-main.pdf

[13] "Geforce® GTX 1060" NVIDIA Corporation, 2017. [Online]. Available: https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1060/

[14] "Geforce® GTX 1070 Family" NVIDIA Corporation, 2017. [Online]. Available: https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1070-ti/

[15] "NVIDIA GeForce GTX 1080," NVIDIA Corporation, 2016. [Online]. Available: https://international.download.nvidia.com/geforce-com/international/pdfs/GeForce_GTX_1080_Whitepaper_FINAL.pdf

Basyir Adam received Bachelor of Engineering (Honours) Electronics Engineering (Computer) at Universiti Teknologi MARA (UiTM), Shah Alam in 2016. Currently doing his Master of Science in Electrical Engineering at UiTM, Shah Alam. His research field is on convolutional neural network.

Fadhlan Hafizhelmi Kamaru Zaman is a Computer Engineering lecturer at UiTM, Shah Alam. He received academic qualification of Master of Science in Communications at International Islamic University Malaysia (IIUM) and Doctor of Philosophy in Engineering at IIUM. His current research area are Video Surveillance, Computer Vision, Machine Learning, Pattern Recognition and Intelligent System.

Ihsan M. Yassin is a Computer Engineering senior lecturer at UiTM, Shah Alam. He received Bachelor of Science in Electrical Engineering, Master of Science in Electrical Engineering and Doctor of Philosophy in Electrical Engineering. He also received Chartered Engineer, IET. His current research area are Web Applications Design and Development, Neural Networks, Database Design, Artificial Intelligence, Fuzzy Logic and System Identification.

Husna Zainol Abidin is a Computer Engineering senior lecturer at UiTM, Shah Alam. She received Bachelor of Engineering in Electrical at University of Wollongong in 2001. Later, she received her Master of Engineering at Universiti Tenaga National (UNITEN) in 2006 and Doctor of Philosophy at UNITEN in 2015. Her current research area are Computer Networking, Teletraffic Monitoring and Wireless Networking.