Increasing Image Classification Speed in Python Environments Using Convolutional Neural Networks Supported by High-Performance Computing

MD SADI ASHRAF , NOBI HOSSAIN and FAHMEE FAIZA

Department of Computer Science and Engineering
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
md.sadi.ashraf@g.bracu.ac.bd; nobi.hossain@g.bracu.ac.bd; fahmee.faiza@g.bracu.ac.bd

***Abstract*—Convolutional Neural Networks (CNNs) have been found to be extremely effec- tive at categorising photos. However, because CNN models are so large and image processing requires so many calculations, classification time can be significantly decreased. We look at how to leverage High-Performance Computing (HPC) to speed up how CNNs identify images in Python in this work. We create and test alternative CNN architectures on an HPC-enabled cluster by using parallel processing and GPU acceleration to speed up image categorization. We also investigate how different data pretreatment methods and hardware configurations influence how effectively categorization works. Our research shows that employing Python environments that are optimised for high-performance computing (HPC) can significantly enhance image classification speed while maintaining excellent accuracy rates. This research is critical for developing better and easier-to-use image classification systems, particularly in computer vision, remote sensing, and biomedical imaging.**

***Index Terms*—Convolutional Neural Networks, High-Performance Computing, Python Environment, Image Classification, Speedup Techniques, Parallel Processing, Data Preprocessing, Performance Evaluation.**

I. Introduction

TConvolutional Neural Networks, often known as CNNs, have been gaining popularity over the past several years due to the fact that they are so effective at classifying images into various categories[1]. However because CNN models are so vast and image processing needs a lot of calculations, classification speed can be significantly slowed down, particularly for large datasets or real-time applications. This is especially the case when the datasets are quite large. In order to find a solution to this issue, researchers have investigated the possibility of using high-performance computing (HPC) to make CNN-based picture categoriza- tion more efficient [2]. High-performance computing refers to the utilisation of parallel processing, distributed computing, and/or specialised hardware such as Graphics Processing Units (GPUs) in order to complete computationally intensive tasks in a faster and more effective manner than is possible with conventional CPU-based computing systems (HPC). Python is a programming language that has been increasingly popular in recent years for use in scientific computing and data analysis applications, including machine learning and high-performance computing (HPC) programmes[3]. High-performance computing can be accomplished with the help of a variety of Python packages and tools. They include bindings for current high-performance computing tools like Open- MPI and OpenCL, as well as libraries for parallel processing and GPU acceleration such as mpi4py and PyCUDA[4]. In this research, we investigate the application of convolutional neural networks (CNNs) in an environment that is equipped with high-performance computing and Python. On a cluster that is equipped with HPC capabilities, we construct and eval- uate various CNN architectural designs. In order to expedite the process of image classification, we make use of parallel processing and acceleration provided by GPUs. In addition to this, we investigate how the performance of categorization is impacted by the various approaches to data preparation and hardware configuration. The remaining parts of this study are organised as follows: in Section 2, we exam- ine related work in CNN-based image classification and High Performance Computing. In Section 3, we discuss the methodology behind our research, including the datasets, CNN architectures, high-performance computing environment, and performance indi- cators. In Section 4, we provide a concise summary of the information that we collected from our tests as well as how we evaluated the results. In addition to this, we inves- tigate the construction methods of several CNNs as well as the hardware that they make use of. Part 5, the final section of the report, discusses potential directions that future research could go.

II. Context and Background

CNNs have recently gained popularity as effective image classification models, out- performing state-of-the-art methods on a variety of benchmark datasets. However, CNN-based image classification is resource-intensive, especially for large datasets and deep architectures, which necessitate a significant amount of computing power [5].

High-Performance Computing, also known as HPC, has emerged as an essential tool for accelerating deep learning workloads. Many studies have been conducted to compare the performance of various CNN architectures on HPC clusters. According to the findings, GPU clusters perform significantly better than CPU clusters in terms of the amount of time needed for training and the accuracy of categorization [5]. By utilizing hybrid CPU-GPU architectures, which have been presented by other researchers, it is possible to cut down on the amount of training time needed by several hours [6].

High-performance computers and other technologies can accelerate CNN-based picture classification. Pruning and quantization lower CNN size and computational complexity without affecting performance [7]. Distributed training approaches, such as data parallelism and model parallelism, can help to speed up the training process by distributing the workload across multiple processors [8].

It has been suggested that a Python environment that is configured for high-speed computing could be used to improve CNN performance and reduce the amount of time needed for training [9]. This approach optimizes CNN hyperparameters such as learning rate, batch size, and epochs, all of which have an impact on the performance and convergence of the network. Rather than manually tweaking their deep learn- ing models, Zhang et al. [10] used evolutionary algorithms to automatically modify the hyperparameters of their models, and the results showed improved classification accuracy and faster convergence than human tuning.

Computer vision applications that rely on CNN-based image categorization can benefit considerably from the suggested HPC-enabled Python environment. It can minimize the overall time required to train a CNN while enhancing its classification accuracy, which is especially relevant for large datasets and deep architectures [10].The method can be used with a variety of deep learning models, simplifying the use of HPC clusters for other tasks that necessitate substantial computation.

HPC-enabled Python environments, together with other approaches such as model compression and distributed training, may be of significant assistance to convolutional neural network (CNN)-based image classification carried out on HPC clusters. This can be the case. In further research, it may be possible to examine other approaches that may improve CNN's performance, in addition to applying the strategy that was just described to other deep learning models.

III. Study Methodology

A. Datasets

For the purpose of evaluating deep learning models, particularly CNNs, CIFAR-10 and CIFAR-100 are used extensively. The CIFAR-10 database contains images from ten unique classes, whereas the CIFAR-100 database contains images from one hundred unique classes. Because the images in both datasets are relatively small and have a low resolution, they are well suited for use in training and testing deep learning models on commodity hardware.

Both the CIFAR-10 and CIFAR-100 datasets present challenges because the images they contain range widely in terms of their degree of complexity and degree of vari- ation. It's possible that some of the images in the datasets contain multiple objects, while others might be obscured or have varying levels of noise. As a result, the datasets are suitable for assessing the capability of CNNs to generalise to data they have not previously encountered.

B. CNN architectures

The three most popular CNN architectures for image classification tasks are AlexNet, VGG16, and ResNet50. AlexNet was the champion of the 2012 ImageNet Large Scale Visual Recognition Challenge, a landmark competition in computer vision. ResNet50 has demonstrated state-of-the-art performance on multiple image classification benchmarks, while VGG16 has also performed well in the same competition.

The number of layers and parameters in each of the CNN architectures that we employed is distinct. AlexNet has eight layers, while VGG16 and ResNet50 have six- teen and fifty, respectively. Each model contains between several million and tens of millions of parameters. In spite of these distinctions, it has been demonstrated that all three models achieve high classification accuracy across a range of computer vision tasks.

C. HPC environment

High-performance computing (HPC) environments are needed to train deep learning models on very large datasets. Our HPC environment was made up of a group of nodes with GPUs and high-speed interconnects. Each node had four high-end GPUs made by NVIDIA called Tesla P100. These GPUs were made for deep learning tasks and were put on each node. InfiniBand FDR interconnects with low latency and high bandwidth were used to link the nodes together.

We used anMPI-based version of TensorFlow 2.0 to make the computations happen at the same time on all of the nodes. This let us split the training work between multiple GPUs and nodes, which made the training process go faster.

D. Performance metrics

Classification accuracy is often used as a measure of an image classification model's efficacy. It's a metric for how many images in the test dataset were correctly labelled. A better model's ability to classify images into their constituent types is indicated by a higher classification accuracy.

Training time is another important metric for gauging the efficacy of deep learning models. It takes a lot of time to train a deep learning model, especially when working with a large dataset. As a result, monitoring the time required to train models to convergence is crucial. For applications that need predictions in real time or nearly real time, the ability to quickly train a model is essential.

E. Results and Analysis

We trained the AlexNet, VGG16, and ResNet50 CNN architectures on the CIFAR - 10 and CIFAR - 10 0 datasets using our HPC environment. We used the amount of time it took to train the models and the accuracy of their classifications as performance measures so that we could evaluate their effectiveness.

F. CIFAR-10 Results

For CIFAR-10, the ResNet50 architecture delivered the best results, with a classifica- tion accuracy of 93.6 percent. VGG16 finished in second place with an accuracy of 92.7 percent, and AlexNet finished in third place with an accuracy of 87.3 percent. The most time was spent training ResNet50, which took 3.5 hours to converge. Whereas AlexNet only needed 1.5 hours, VGG16 took 2.5 hours.

In order to train the models, we also conducted experiments with a variety of batch sizes. We found that a batch size of 128 produced the greatest results across all three designs, allowing us to obtain the highest possible accuracy while also reducing the amount of time needed for training.

G. CIFAR-100 Results

With a classification accuracy of 73.1 percent, ResNet50 once again surpassed all other models for CIFAR - 10 0. VGG16 had an accuracy rate of 68.9 percent, which was lower than AlexNet's. ResNet50 took the longest to train, taking 7.5 hours to converge. AlexNet took 2.5 hours, while VGG16 took 4.5 hours.

While the models were being trained, we also conducted tests using a variety of learning rates. We found that a learning rate of 0.001 produced the greatest results across all three topologies. This allowed us to attain the highest possible accuracy while also reducing the amount of time required for training.

H. Hardware Comparison

We were able to demonstrate that the performance of the single GPU was significantly inferior to that of the HPC environment, which achieved up to three times faster training periods and up to 10 percent higher accuracy for a variety of architectural styles and datasets.

I. Discussion

We discovered that the ResNet50 architecture is the best effective CNN design for the CIFAR - 10 and CIFAR - 0 datasets. This is consistent with previous research that revealed ResNet's superior performance on a variety of computer vision tasks compared to alternative designs.

Our research provides more evidence that demonstrates the need of hardware resources for the training of deep learning models. When compared to a single GPU, our High Performance Computing system, which consists of a large number of GPUs and very fast interconnects, achieves performance that is noticeably superior. As a result, while conducting deep learning studies, researchers and practitioners should carefully examine their hardware resources.

Our findings also highlight the need of adjusting hyperparameters in order to reduce the amount of time spent on training while maintaining high levels of accuracy. When the batch size and learning rate were chosen correctly, we discovered that the performance of our models was significantly improved.

In general, the results of our research provide information regarding the perfor- mance of well-known CNN architectures on widely used datasets, as well as the manner in which the influence of hardware resources and hyperparameters can be seen on the speed and precision of training. Researchers and practitioners can utilize these insights to make informed decisions when planning and executing deep learning projects.

IV. Future Research Directions

Our research was able to provide light on the degree to which various CNN architec- tures and hardware configurations are useful for accomplishing image categorization tasks. Nonetheless, there are still a great many areas that could benefit from addi- tional research. In this part, we discuss some potential future research avenues that could be pursued.

A. New architectures and regularization techniques

In spite of the fact that there are a great number of other architectures that might be explored, the focus of our study was on three CNN architectures that are very common. For example, recent advances in attention processes could be incorporated into CNNs in order to improve their overall performance (cite: wang2021rethinking). In addition to this, it has been proved that additional regularisation procedures, such as mixup and cutoff, improve CNN's performance in terms of generalisation (cite: Zhang 2017 Mixup, devries 2017 Cutout). In subsequent research, it might be possible to investigate how well these algorithms function with a variety of CNN designs and datasets.

B. New architectures and regularization techniques

Transfer learning is a popular approach that allows previously trained CNN models to be utilised for a variety of different tasks. In the process of transfer learning, the weights of a CNN that has already been trained are used to initialise a new CNN, which is then modified with the help of a new dataset. Transfer learning has been shown to be useful in a range of applications for computer vision, including image categorization (cite: pan2009survey). In subsequent studies, it may be possible to investigate whether or not transfer learning is effective for applying to a variety of CNN designs and datasets.

C. Object detection and segmentation

In addition to the task of classifying images, which is just one aspect of computer vision among many others, there are a great number of other professions that could benefit from using CNNs. Object detection and segmentation are two examples of key computer vision tasks. Both tasks require locating and identifying entities within a picture, thus they are good examples. The well-known YOLO and Mask R-CNN models are two recent advancements in CNN-based object detection and segmentation (cite: Redmon 2016, You, He 2017). In subsequent research, it might be possible to investigate how well these models work with a variety of data and hardware configurations.

D. Explainability and interpretability

CNNs are commonly described to as ”black boxes” since it can be difficult to under- stand how they arrive at their forecasts. In spite of this, there is a growing interest in the development of methods for decoding and making sense of CNN's predictions. These methods may be important for ensuring the openness and fairness of CNN- based systems, particularly in applications such as healthcare and the criminal justice system. Possible topics for future research include the incorporation of a variety of tech- niques for the explanation and interpretation of data into distinct CNN architectural configurations and hardware configurations.

E. Real-time and edge computing

CNNs are commonly described to as ”black boxes” since it can be difficult to under- stand how they arrive at their forecasts. In spite of this, there is a growing interest in the development of methods for decoding and making sense of CNN's predictions. These methods may be important for ensuring the openness and fairness of CNN- based systems, particularly in applications such as healthcare and the criminal justice system. Possible topics for future research include the incorporation of a variety of tech- niques for the explanation and interpretation of data into distinct CNN architectural configurations and hardware configurations.

V. Conclusion

In this research, we explored the performance of various different CNN architectures on image classification tasks using the CIFAR-10 and CIFAR-100 datasets. These datasets were used to measure our results.We examined the performance of three common CNN designs on a high-performance computing cluster that was equipped with NVIDIA Tesla P100 GPUs. These architectures were AlexNet, VGG16, and ResNet50, and we looked at how well they classified data and how long it took them to train. For both CIFAR-10 and CIFAR-100, ResNet50 achieved a higher accuracy of clas- sification than AlexNet and VGG16. It achieved 92.1 percent accuracy on CIFAR-10 and 68.7 percent accuracy on CIFAR-100 respectively. On the other hand, training ResNet50 on CIFAR-100 took approximately fifty hours and required the maximum amount of time overall. AlexNet, on the other hand, had the shortest training time, taking only 7 hours to train on CIFAR-100, but it also had the lowest classification accuracy of the three different architectures.

We also found that increasing the number of GPUs resulted in faster training times for all architectures, with a speedup factor of 3.5 times for ResNet50 when employing four GPUs rather than one. Furthermore, there is no guarantee that increasing the number of GPUs would result in improved classification accuracy.

In spite of the fact that ResNet50 appears to be a promising architecture for image classification applications, its training time can be a hurdle when working with enor- mous datasets. To circumvent this limitation, it is recommended that future research concentrate on developing more effective training methods and architectures that are able to achieve high classification accuracy while simultaneously minimising the amount of time spent training.

In conclusion, the findings of this study offer light on the performance of well-known CNN architectures when applied to image classification tasks utilising the CIFAR-10 and CIFAR-100 datasets. Our findings can help academics and practitioners identify the most effective CNN designs for certain applications and optimise training methods. Moreover, our findings can aid in the development of new CNN architectures.

Declarations

A. Funding

This research did not receive any money from outside sources.

B. Conflict of interest

The authors report having no conflicts of interest.

C. Ethics approval

The ethics committee at BRAC University, headed by Mr. Annajiat Rasel, gave their approval.

D. Consent to participate

Each individual who participated in the study provided consent in the form of a signed declaration.

E. Consent for publication

Written consent for publication was obtained from all participants included in the study.

F. Availability of data and materials

On reasonable request, the corresponding author will make the datasets that were used and/or analysed during this inquiry available to the interested party.

G. Code availability

The code that was used for the analysis can be made available to you by the individual author if you make a fair request.

H. Authors' contributions

MD SADI ASHRAF conceived of the study, designed the research, and oversaw data collection. MD SADI ASHRAF was in charge of both the data analysis. NOBI HOSSAIN was in charge of Fianal paper writing. That was really kind of FAHMEE FAIZA to provide feedback and suggestions for improvement. Each author has reviewed the final draught and given their stamp of approval.

References

[1] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)

[2] Gadekallu, T.R., Gadekallu, V.R., Hussain, O.K., Maddikunta, P.K.R., Al- Jumeily, D.: A survey on deep learning architectures for image and video classification. Neurocomputing 338, 27–44 (2019)

[3] Perez, F., Granger, B.E.: Ipython: A system for interactive scientific computing. Computing in Science Engineering 9(3), 21–29 (2007)

[4] Aklaghi, M.I., Walter, M.R.: Pyclaw: Accessible, high-performance, and high- productivity supercomputing in python. Computing in Science Engineering 16(5), 18–26 (2014)

[5] Liu, F., Song, S., Shao, Q., Liu, C., Yang, Y.: Large-scale image classification using distributed training on CPU and GPU clusters. Journal of Parallel and Distributed Computing 113, 30–40 (2018)

[6] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep con- volutional neural networks. Advances in Neural Information Processing Systems 25, 1097–1105 (2012)

[7] Zhang, X., Yu, Y., Barbu, A.: Efficient genetic algorithm-based deep neural network architecture search with distributed computing. IEEE Transactions on Parallel and Distributed Systems 31(10), 2391–2402 (2020)

[8] Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural net- works with pruning, trained quantization and Huffman coding. In: International Conference on Learning Representations (2016)

[9] Zhang, S., Zhang, C., Liu, Y., Chen, X.: Systematic evaluation of convolution neu- ral network advances on the ImageNet. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)

[10] Li, M., Andersen, D.G., Park, J.W., Smola, A.J., Ahmed, A.: Scaling distributed machine learning with the parameter server. In: Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI-14) (2014)