ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7. Available from http://www.i6doc.com/en/.

Data-Efficient Training of High-Resolution Images in Medical Domain

Amey Pandit*, Kushagra Mahajan*, Shruti Kunde, Monika Sharma, Rekha Singhal and Lovekesh Vig

TCS Research, India

Abstract. The ability of Graphical Processor Units (GPUs) to quickly train data- and compute-intensive deep networks has led to rapid advancements across diverse domains such as robotics, medical imaging and autonomous driving. However, memory constraints with GPU-based training for memory-intensive deep networks have forced researchers to adopt various workarounds: 1) resize the input image, 2) divide input image into smaller patches, or use smaller batch-sizes in order to fit both the model and batch training data into GPU memory.While these alternatives perform well when dealing with natural images, they suffer from 1) loss of high-resolution information, 2) loss of global context and 3) sub-optimal batch sizes. Such issues will likely to become more pressing for domains like medical imaging, where data is scarce and images are often of very high resolution with subtle features. Therefore, in this paper, we demonstrate that training can be made more data-efficient by using a distributed training setup with high-resolution images and larger effective batch sizes, with batches being distributed across multiple nodes. The distributed GPU training framework, which partitions the data and only shares model parameters across different GPUs, gets around the memory constraints of single GPU training. We conduct a study in which experiments are performed for different image resolutions (ranging from 112×112 to 1024×1024) and different number of images per class to determine the effect of image resolutions on network performance. We illustrate our findings on two medical imaging datasets namely, SD-198 skin-lesion and NIH Chest X-rays.

1 Introduction and Motivation

The recent breakthroughs in deep learning have made it possible to create automated practical solutions to challenging medical diagnosis problems [1], [2]. Deep neural networks require a large amount of annotated data for training to avoid over-fitting, which is often a concern in medical imaging datasets where few annotated images but of high-resolution are available. However, limited GPU-compute forces us to work with low-resolution images hence, high-resolution images are down-sampled for training purposes [3]. This compromises the network performance as size reduction incurs a significant loss of crucial information. Furthermore, the batch size has to be reduced for high resolution images to fit the deep model and batch training data into GPU memory. Thus, there is a trade-off as higher batch size allows better gradients to be computed with regard to the loss function [4]. In addition, sharing of sensitive patient data coming from different sources is a key concern, and a deterrent for training deep learning models that require large amounts of training data. Therefore, to enable training with

*Equal contribution

large effective batch sizes and avoid sharing of sensitive patient data, we resort to distributed training of high-resolution images in the medical domain.

In this paper, we intend to address fundamental challenges of low-data, high resolution images and optimal effective batch-size distributed across worker nodes, in the training of deep neural networks on medical imaging data for automated diagnosis. We conduct experiments to probe the following: 1) Can fewer high-resolution images be used to train, rather than a large number of low-resolution images, and yet achieve comparable / better accuracy? 2) What are the advantages of training high-resolution images in a distributed GPU setup? 3) Can we select an optimal image resolution for training a deep network which provides us with a better network performance for a medical diagnosis task?

With the above motivation, we conduct experiments on two publicly available medical imaging datasets: SD-198 skin lesion [5]; and NIH Chest X-rays [6]; to study the impact of different image resolutions (from 112×112 to 1024×1024), and varying number of images per class, on deep network performance in two settings: single and distributed GPU training setup.

2 Related Work

High-resolution images are desirable [7], [8] as they yield better network performance and provide valuable insights into identifying a particular diseases. There have been few attempts earlier to alleviate batch size problem in such scenarios. Haruki et al. [9] uses data-swapping to enlarge the effective GPU memory for training a 3D medical image segmentation network, without breaking an image into small patches. In [3], the authors implemented spatial partitioning in a Mesh-TensorFlow framework. The study in [7] concludes that increasing image resolution for CNN training often has a trade-off with the maximum possible batch size, but this is amenable to optimization for maximization of neural network performance. In essence, earlier works have mostly focused on devising ways to train deep networks on high-resolution images with effective GPU utilization. In our paper, we are conducting a study to find the effect of different image resolutions on network performance and determining if high resolution images in the medical domain are more beneficial for the diagnosis.

We resort to distributed training of high-resolution images in the medical domain to handle larger effective batch sizes. While distributed GPU training has shown impressive improvements in [10], there exists limited body of work on distributed training of deep networks in healthcare [11]. In [12], authors introduced a scalable intuitive deep learning toolkit called R2D2 for medical image segmentation by offering novel distributed versions of two well-known and widely used CNN segmentation architectures. [13] described a system that leverages Virtual Imaging Platform (VIP) and Distributed Infrastructure with Remote Agent Control (DIRAC) enabling researchers to use distributed computing resources with GPUs for their specific medical imaging applications. We use the open-source distributed training platform called Horovod [14] to conduct experiments in a distributed-GPU setup that allows for a larger effective batch size of images, and varying learning rates.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

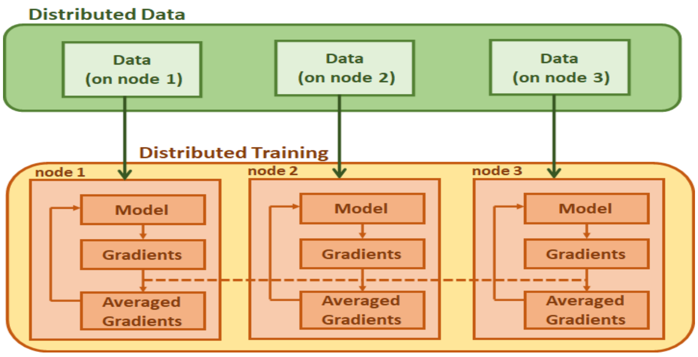

Fig. 1: Distributed GPU training of high resolution medical images using Horovod [14].

Table 1: Summary of our experiments performed on two datasets.

Dataset	No. if classes	Image Sizes	CNN Arch	Training Images per Class
SD-198 [5]	10	112×112, 224×224, 448×448, 512×512, 600×600, 800×800, 1024×1024	VGG16, ResNet50	10, 20, 30
NIH [6]	4	224×224, 448×448, 512×512, 1024×1024	DenseNet-169	200, 400, 600

3 Distributed Training Paradigm

We propose a data-efficient solution for training medical imaging data in a distributed setup. As shown in Figure 1, data is distributed across multiple nodes, each having a copy of model. Every node independently loads data in batches and trains a model and determines gradients for each batch. Thus, each node looks at only a subset of data at any given point during training and subsequently all the gradients across all the nodes are averaged. Various approaches (e.g. all-reduce[14]) can be employed to gather and average the gradients. The model is updated using the gradients and each node then has a copy of the newly updated model. As each node trains on a subset of the data, only model parameters are shared across nodes. Hence, no sensitive patient data is shared between nodes, especially important in the medical imaging domain.We showcase the efficacy of our approach with image datasets of skin lesions and chest X-rays.

4 Experimental Results and Discussions

Environment Setup: We conducted all experiments on an NVIDIA[1] DGX-2 32 GB (16xTesla V100) and NVIDIA A100 Tensor Core GPU running Ubuntu 18.04.4. The training of deep models was carried out using Keras 2.4.0 with TensorFlow 2.3.1 as the backend. A GPU-enabled tensorflow docker file from GitHub which includes TensorFlow and CUDA/cuDNN dependencies was utilized. For the distributed training setup, we used Horovod, an open source distributed training platform that has proven to be

[1] We thank NVIDIA for providing us access to V100 and A100 GPUs for carrying out our experiments.

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7.
Available from http://www.i6doc.com/en/.

beneficial in distributed-GPU training [14]. The evaluation was mainly done on V100 or A100 GPUs using 2 or 4 workers.

Experiment Details: The details of our experiment setup are represented in Table 1. In our experiments with the SD-198 dataset, we use 10 classes with maximum amount of data i.e., 30 images each for training and testing. For experiments with the NIH dataset, since chest X-ray images in the dataset are very complex and contain a lot of variation in illumination as well as noise (e.g. pipes and other medical equipment), we found heuristically that NIH dataset begins to learn discriminating features resulting in adequate accuracy on using minimum of 200 images per class. Therefore, we choose classes from NIH dataset such that each of them contains more than 1000 images per class. Accordingly, we select 4 classes for our experiments. We create 3 different training sets having 200, 400, 600 images per class from these 4 selected classes. We utilize the entire test-set from these selected classes of NIH dataset.

Table 2: Pre-trained networks on SD-198 and NIH datasets.

Dataset: SD-198. Networks: Pre-trained VGG16 and ResNet50						
Image Size	**#Images per class**	**Serial**		**Distributed**		
		Num nodes	**Accuracy(%)** VGG‖Resnet	**Num nodes** VGG‖Resnet	**Learning rate** VGG‖Resnet	**Accuracy(%)** VGG‖Resnet
3*112 × 112	**10**	1	42.90‖48.63	4‖4	8.00e-5‖8.00$e-5$	**51.0**‖**51.0**
	30	1	62.47‖65.95	4‖4	8.00e-5‖4.00$e-5$	**66.8**‖**67.78**
3*224 × 224	**10**	1	**47.12**‖57.38	4‖4	4.00e-5‖4.00$e-5$	45.0‖**59.06**
	30	1	65.76‖77.64	4‖4	4.00e-5‖8.00$e-5$	**68.1**‖**78.52**
3*448 × 448	**10**	1	40.68‖63.42	4‖4	8.00e-5‖8.00$e-5$	**42.3**‖**65.43**
	30	1	59.40‖78.19	4‖4	4.00e-5‖4.00$e-5$	**69.1**‖**79.19**
3*512 × 512	**10**	1	43.39‖63.97	4‖4	4.00e-5‖4.00$e-5$	**46.6**‖**65.1**
	30	1	63.85‖79.90	4‖4	4.00e-5‖8.00$e-5$	**68.5**‖**81.87**
3*1024 × 1024	**10**	1	38.85‖**59.40**	4‖4	4.00e-5‖4.00$e-5$	**45.0**‖56.71
	30	1	60.13‖73.40	4‖4	4.00e-5‖8.00$e-5$	**66.4**‖**82.88**
Dataset: NIH. Network: Densenet169						
3*224 × 224	**200**	1	57.73	2	8.00e-6	**58.9**
	400	1	59.99	2	8.00e-6	**61.2**
	600	1	62.78	4	1.60e-5	**63.0**
3*448 × 448	**200**	1	59.67	2	1.60e-5	**61.6**
	400	1	63.25	2	8.00e-6	**63.66**
	600	1	63.61	8	1.60e-5	**63.84**
3*512 × 512	**200**	1	62.66	3	1.60e-5	**63.4**
	400	1	63.7	2	1.60e-5	**64.79**
	600	1	66.56	6	6.4e-5	**67**
3*1024 × 1024	**200**	1	59.40	4	8.00e-6	**59.8**
	400	1	62.17	4	8.00e-6	**63.3**
	600	1	66.74	4	8.00e-6	**67.1**

Results and Discussions: Now, we present results and insights obtained from our experiments. In the serial setting, the deep network is trained on a single GPU, and hence there is a limitation on the amount of image data that can be passed in a batch. We tried various convolutional neural network (CNN) architectures, as shown in Table 1, and observed that for the SD-198 dataset, the architectures that give the best performance are VGG16 and ResNet50. DenseNet-169 performs better than other standard architectures for the NIH Chest X-rays dataset. In the serial case, owing to GPU memory limitation, we use a batch size of 5 for smaller image resolutions and 4 for 1024 image sizes. We heuristically determine the best learning rate to be $1e-5$ for both the datasets.

The results of our experiments on both the datasets are shown in Table 2. We ob-

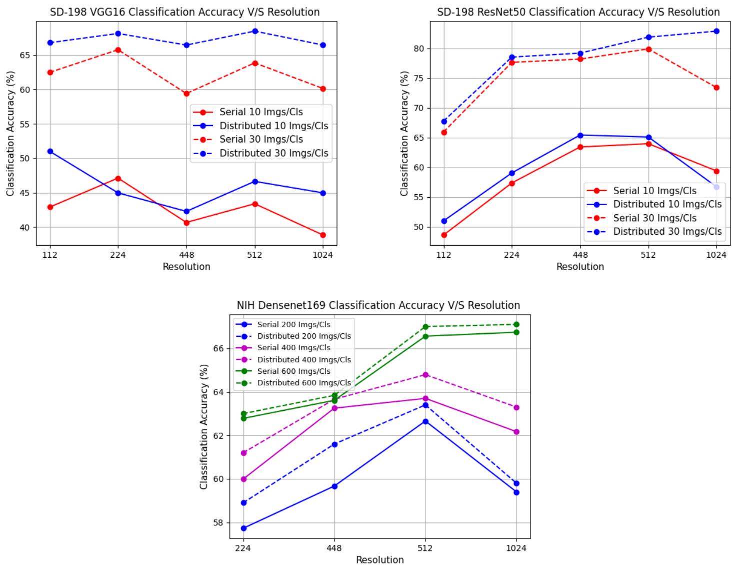

Fig. 2: Graphs showing Image-resolutions vs accuracy for single vs distributed GPU-setup

serve that as the number of training images per class increases, the classification accuracy improves, which is expected since more data results in better generalization of the network. Another interesting insight is that, with the increase in input image size, the network performance improves initially, but begins to degrade once the image size goes beyond a certain threshold (Refer Table 2, SD-198 VGG16 and NIH for image size 1024 and 512). Since the standard CNN architectures generally used for classification tasks are designed with the default input size of 224×224, their smaller-kernel architectures are not suitable when the image sizes get extremely large (i.e. spatial neighbourhoods become extremely fine-grained, so cannot be accommodated in the kernel).

Further, we observe that in a distributed scenario, every image size has a corresponding ideal learning rate that attains the best network accuracy. This is reflected in both datasets. Also distributing tasks across multiple GPUs enables a larger effective batch size which boosts the accuracy, as is evident in Table 2. We observe similar findings in multiple experiments by varying image dimensions in both the datasets.

Another interesting observation is that for different image resolutions given the model is kept constant, there is an optimal resolution at which the model will deliver the best performance on the current task. Any resolution that is higher or lower, will affect performance adversely (Refer Figure 2.) Besides, as the number of images-per-class increases, the difference in the accuracy between serial and distributed training becomes more significant (Table 2 SD198-ResNet50), with distributed training clearly outperforming serial training of the model.

We demonstrate that distributed training usually attains a higher accuracy as compared to serial training, with a few exceptions. The gradients are evaluated across multiple nodes in parallel as the effective batch size increases thus resulting in a higher

ESANN 2021 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 6-8 October 2021, i6doc.com publ., ISBN 978287587082-7. Available from http://www.i6doc.com/en/.

accuracy. Thus, we believe that a distributed training setup is particularly beneficial in the medical imaging domain, where most of the images are of a very high resolution and a distributed training setup, facilitates training in parallel across multiple nodes.

5 Conclusions and Future Work

In this paper, we have demonstrated how training can be made more data-efficient by using distributed GPU training on high resolution images. Most images in the medical domain have high resolutions, rendering it infeasible to train with large effective batch sizes, owing to memory constraints. This severely limits the accuracy of the trained network. A distributed GPU-setup overcomes these challenges enabling us to distribute the data across multiple nodes, resulting in a better accuracy as compared to a serial setting. We demonstrate our experimental results on two datasets from the medical domain. As a part of the future work, we plan to expand our study on the efficacy of a distributed training approach to other unexplored fields of the medical imaging domain.

References

[1] Ozan Oktay et al. Attention u-net: Learning where to look for the pancreas. *CoRR*, abs/1804.03999, 2018. 1

[2] Yi et al. Generative adversarial network in medical imaging: A review. *Medical Image Analysis*, 58:101552, 2019. 1

[3] Hou et al. High resolution medical image analysis with spatial partitioning, 2019. 1, 2

[4] Sabottke et al. The effect of image resolution on deep learning in radiography. *Radiology: Artificial Intelligence*, 2(1):e190015, 2020. 1

[5] Xiaoxiao et al. A benchmark for automatic visual classification of clinical skin disease images. In *European Conference on Computer Vision*, pages 206–222. Springer, 2016. 2, 3

[6] Rajpurkar et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. 2017. 2, 3

[7] Munadi et al. Image enhancement for tuberculosis detection using deep learning. *IEEE Access*, 8:217897–217907, 2020. 2

[8] Isaac et al. Super resolution techniques for medical image processing. In *2015 International Conference on Technologies for Sustainable Development (ICTSD)*, pages 1–6, 2015. 2

[9] Haruki et al. High resolution medical image segmentation using data-swapping method. In *MICCAI*, pages 238–246, 2019. 2

[10] Mahon et al. Performance analysis of distributed and scalable deep learning. In *2020 IEEE*, pages 760–766. IEEE, 2020. 2

[11] NVIDIA. Nvidia clara imaging, 2021. Last accessed 18 August 2021. 2

[12] Soulaimane et al Guedria. R2D2: A scalable deep learning toolkit for medical image segmentation. In *Journal of Software: Practice and Experience*, August 2020. 2

[13] Camarasu-Pop et al. Exploiting GPUs on distributed infrastructures for medical imaging applications with VIP and DIRAC. In *42nd international convention on information and communication technology, electronics and microelectronics*, pages 190–195, May 2019. 2

[14] Sergeev et al. Horovod: fast and easy distributed deep learning in tensorflow. *arXiv preprint arXiv:1802.05799*, 2018. 2, 3, 4