Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Checxray: A Saas Application for Chest X-Ray Diagnosis

Vishnuvardhan Chappidi
Student (B.E.),
Department of Computer Engineering,
Pune, India

Mohd. Arbaaz Shaikh
Student (B.E.),
Department of Computer Engineering,
Pune, India

Shubham Bhan
Student (B.E.),
Department of Computer Engineering, Pune, India

Hrishikesh Mane
Student (B.E.),
Department of Computer Engineering,
Pune, India

Dr. Rubeena Khan
Professor,
Department of Computer Engineering,
Pune, India

***Abstract*— Training a Deep learning model is a highly computational demanding task and requires a high-end graphics card for parallelism in training a Deep Convolutional Neural Network for numerical calculation. These production level cards are not easily accessible to everyone and are extremely costly. The paper discusses a distributed training strategy used for training a deep convolutional neural network on multiple consumer graphics cards along with the deployment architecture for inference on a SaaS application.**

Keywords*—*Deep Learning, Deep Convolutional Neural Network, Medical Imagining, Healthcare.

I. INTRODUCTION

The field of medical imaging is a highly competitive field for Computer Assisted Diagnostics due to recent breakthroughs in the field of deep learning and computer vision; especially because of data augmentation techniques to solve the problem of fewer data and transfer learning, where previous knowledge of the neural network is leveraged in the form of neural network weights as initial weights for the training of an expert system.

Even though transfer learning helps save the training time and the computation on a deep neural network, significant computational resources are required for fine-tuning the model to yield higher accuracy. Training a deep convolutional neural network on a large dataset requires a high-end graphics card with more other resources like faster memory and hard drives for low latency. This problem can be easily solved by cloud platforms but the cost of such resources is extremely high. The other solution is training of the neural network on a multiple distributed low powered card.

The trained model can be exported once obtained the desired result. This model can be used for inference on the client-side for any unknown input image of the domain. The model cannot be shipped with the application because of large network overhead and also because of difficulty in continuous integration and continuous deployment. Thus, making a continuous serving model is a necessity for faster inference.

II. PROBLEM STATEMENT

The problem is to solve the repetitive task of the radiologist of classifying the Chest X-rays not only as malignant or benign but also specifying the thoracic pathology inferred from the given X-ray as an input along with confidence and a heatmap. The given Chest X-ray may have two or more pathologies at the same time; therefore, it is the task of multi-class multi-label classification.

III. DATASET

The Dataset used is a widely known, publicly available dataset of more than 100,000 high resolution images frontal Chest X-rays by NIH Clinical Centre. The dataset is gathered from the scans of more than 30,000 anonymous patients. The dataset is labelled for 14 different thoracic pathologies annotated by the professional radiologists.

The NIH Chest X-ray 14 dataset has 6 more classes and more images than in recent work [1]. The large and varied dataset represents quite a realistic representation of the distribution of the Chest X-rays, which in case holds a higher chance of yielding realistic diagnostic results. The dataset has 112,120 images of frontal Chest X-rays of 30,805 patients. The labelled classes in the dataset include Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural thickening, Cardiomegaly, Nodule, Mass and Hernia. Table 1. Shows the distribution of images per class.

Class	Number of Images
Atelectasis	11,535
Cardiomegaly	2,772
Effusion	13,307
Infiltration	19,871
Mass	5,746
Nodule	6,323
Pneumonia	1,353
Pneumothorax	5,298
Consolidation	4,667
Edema	2,303
Emphysema	2,516
Fibrosis	1,686
Pleural Thickening	3,385
Hernia	227

Table 1. Distributed of images per class.

IV. EXPERIMENTS

The Convolutional Neural Network architecture went through a number of experiments with many state of the art convolutional neural networks as the base model with imagenet weights.

Table 2. describes the AUROC score for classification on test dataset for each model. DenseNet121 of all the experiments yielded the best results. The hyperparameters, loss function and the optimizer for DenseNet121 are the same as mentioned in [2]. The model has an input shape of (224, 224, 3) with Densenet121 as base model and the output layer as a sigmoid activation function for 14 classes which outputs a one-dimensional vector for the confidence on each class.

Deep convolutional neural networks have been proven in the past to understand complex features in images, making medical imaging an ideal choice for the same. Deeper the model, greater is the understanding of the features. But deeper models bring the problem of vanishing gradients. The gradients get lost in the mid-way while back- propagating and thus making the later layer not fine-tune. DenseNet solved this problem with skip connections making gradients reach to later layers too. Making DenseNet a great choice for building deep models for learning of complex features in the field of medical imaging.

Class	**DenseNet169**	**DenseNet201**	**DenseNet121**
Atelectasis	0. 7941	0. 8241	0.8263
Cardiomegaly	0. 8431	0. 8131	0.8732
Effusion	0. 8401	0. 8201	0.8939
Infiltration	0. 7221	0. 7338	0.7325
Mass	0. 8353	0. 8567	0.8479
Nodule	0. 7311	0. 7533	0.7633
Pneumonia	0. 7432	0. 7321	0.7784
Pneumothorax	0. 8834	0. 8542	0.9084
Consolidation	0. 7226	0. 7061	0.7406
Edema	0. 8245	0. 8732	0.8847
Emphysema	0. 8973	0. 8635	0.9595
Fibrosis	0. 7972	0. 8002	0.8003
Pleural Thickening	0. 7926	0. 7862	0.8092
Hernia	0. 7963	0. 8012	0.8295

Table 2. Experiment results

V. TRAINING STRATEGY

Training of the deep convolutional neural network is itself a complex task and requires a high computational resource. The experiments range from the training of the model of 16 layers to 201 on more than 100,000 images. This highly computational task can be achieved by distributed training of the model on multiple low power GPUs on multiple nodes at disposal, to save the cost of training on high-end hardware

The experiments were conducted with multi-node training strategy. The strategy implements synchronous distributed training across multiple nodes, that is all the modes train over different parts of input data in sync, and aggregate the gradients at each step during training, with each node having a single or multiple GPUs. The key difference between multiple GPU training and multi node-training is that multi-node training is done on multiple GPUs on multiple systems in a network than multiple GPUs on a single system on a single system. The multi-node uses an all-reduced communication method to keep variables in sync with each other. Figure 1. Describes the basic architecture of the multi-node training strategy.

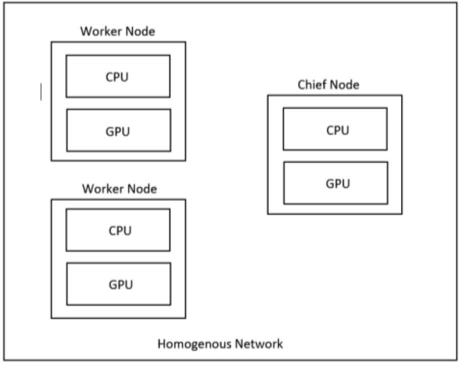

Fig 1. Distributed training architecture

The chief node is responsible for additional tasks like saving logs and checkpoints. This training strategy enables the building of flexible GPU clusters for faster and parallel training. The communication can be implemented using ring-based collectives using remote procedure calls as the communication layer or by NVIDIA’s Collective Communications Library (NCCL) [10]. Figure 2. and 3. shows the plot for loss function and learning rate, respectively.

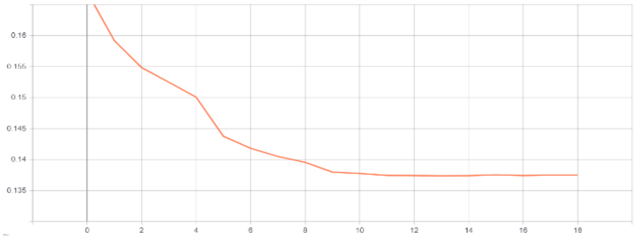

Fig 2. Distributed training architecture

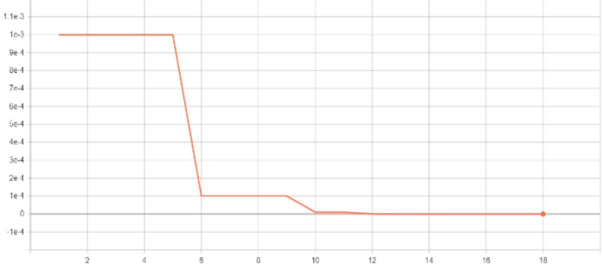

Fig 3. Distributed training architecture

Classes	**Wang et al.** [1]	**Yao et al.** [8]	**CheXNet** [6]	**Ours**
Atelectasis	0.716	0.772	0.8094	0.8263
Cardiomegaly	0.807	0.904	0.9248	0.8732
Effusion	0.784	0.859	0.8638	0.8939
Infiltration	0.609	0.695	0.7345	0.7325
Mass	0.706	0.792	0.8676	0.8479
Nodule	0.671	0.717	0.7802	0.7633
Pneumonia	0.633	0.713	0.7680	0.7784
Pneumothorax	0.806	0.841	0.8887	0.9084
Consolidation	0.708	0.788	0.7901	0.7406
Edema	0.835	0.882	0.8878	0.8847
Emphysema	0.815	0.829	0.9371	0.9595
Fibrosis	0.769	0.767	0.8047	0.8003
Pleural Thickening	0.708	0.765	0.8062	0.8092
Hernia	0.767	0.914	0.9164	0.8295

Table 3. Result Comapison

VI. DEPLOYMENT

The goal of serving system of the model for inference is a flexible and high-performance serving system with the support of continuous integration and continuous deplanement. The Deployment strategy has a pipeline that supports continuous integration and continuous deplanement as well as model versioning, where different versions of models will be available for making API requests from the client-side. This model serving strategy is highly scalable, both horizontally and vertically on one or many GPU cluster with the container management system. The serving system can handle both pre-processed as well as raw data for inference by having different API header for clients to query and scripts for pre-processing on the server. Figure 4. Describes the basic architecture of the deployment strategy.

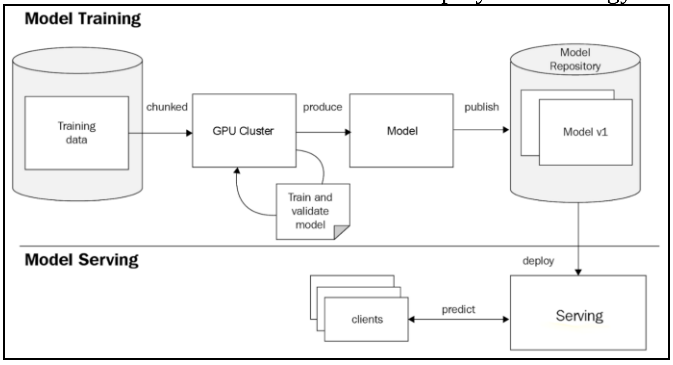

Fig 4. Deployment Stratergy

VII. SAAS APPLICATION

The model on the serving system responds to the requests by a command-line interface and HTTP request. The user-friendly SaaS application, built around the serving system architecture make the request to the latest model deployed in the model repository in the serving system. The request made by the application is on the pre-processed input of the user to save a large network overhead. The inference is then acquired on the application, that is the client-side. The Inference includes the diagnosed disease along with a heatmap of the Chest X-ray. A report is then generated on the client-side itself based on the inference. Figure 5. Gives a brief of the working of the application.

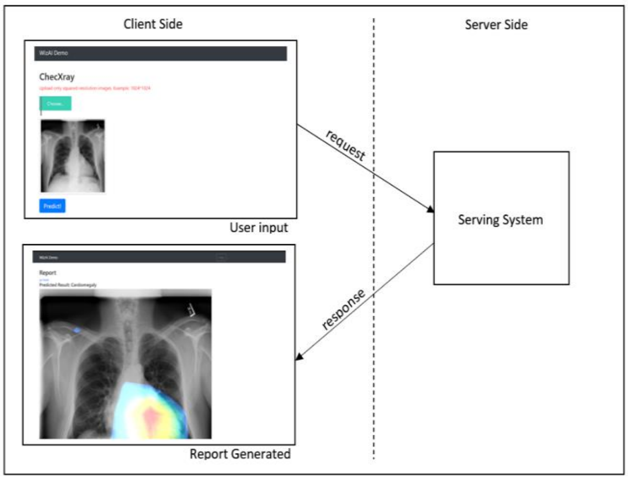

Fig 5. Distributed training architecture

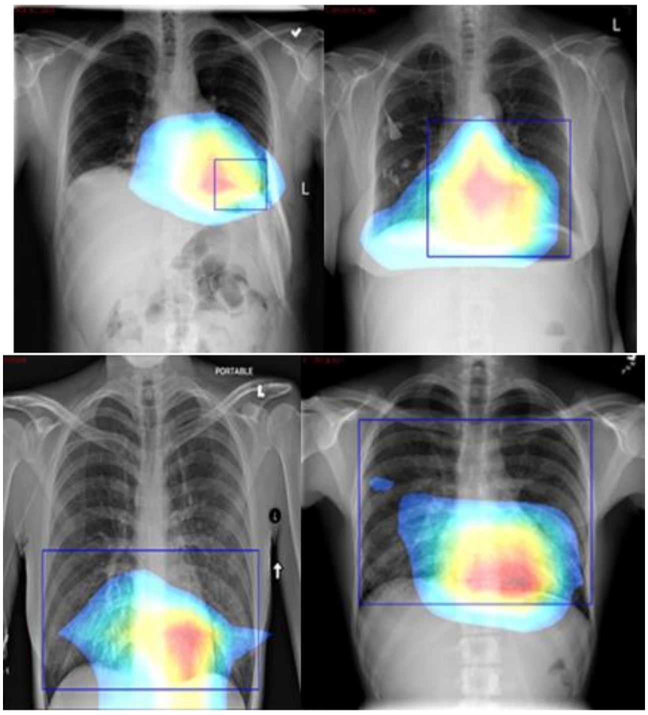

Fig 6. Results from train dataset

VIII. CONCLUSION

The repetitive task of radiologists to identify pathologies in Chest X-rays can be automated which can achieve high accuracy. The training strategy of distributed computing on multiple nodes with low power GPUs can be used for experimenting to reduce the production cost. With the highly scalable deployment of the serving system, inference can be obtained in batch size which results in fast diagnosis of diseases of multiple X-rays.

IX. REFERENCES

[1] Wang, Xiaosong, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097-2106. 2017.

[2] H. Mane, P. Ghorpade and V. Bahel, "Computational Intelligence Based Model Detection of Disease using Chest Radiographs," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-5, doi: 10.1109/ic-ETITE47903.2020.484.

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[4] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[5] Huang, Gao, Shichen Liu, Laurens Van der Maaten, and Kilian Q. Weinberger. "Condensenet: An efficient densenet using learned group convolutions." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2752-2761. 2018.

[6] Rajpurkar, Pranav, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning." *arXiv preprint arXiv:1711.05225* (2017).

[7] Shin, Hoo-Chang, Kirk Roberts, Le Lu, Dina Demner-Fushman, Jianhua Yao, and Ronald M. Summers. "Learning to read chest x-rays: Recurrent neural cascade model for automated image annotation."

In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2497-2506. 2016.

[8] Yao, Li, Eric Poblenz, Dmitry Dagunts, Ben Covington, Devon Bernard, and Kevin Lyman. "Learning to diagnose from scratch by exploiting dependencies among labels." *arXiv preprint arXiv:1710.10501* (2017).

[9] Maji, Kamal Jyoti, Anil Kumar Dikshit, and Ashok Deshpande. "Disability-adjusted life years and economic cost assessment of the health effects related to PM 2.5 and PM 10 pollution in Mumbai and Delhi, in India from 1991 to 2015." *Environmental Science and Pollution Research* 24, no. 5 (2017): 4709-4730.

[10] https://developer.nvidia.com/nccl