Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 18s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Enhancing Large-Scale Medical Image Processing in HPC Using Distributed Convolutional Neural Networks

Abhishek Gandhar[1], S. B. Kumar[2], Shashi Gandhar[3], Prakhar Priyadarshi[4]
Arvind Rehalia[5], Mohit Tiwari[6]

[1]Professor, Electrical and Electronics Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India, Email:- abhishek.gandhar@bharatividyapeeth.edu

[2]Associate Professor, Electronics and Communication Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India, Email:- shashi.kumar@bharatividyapeeth.edu

[3]Associate Professor, Electrical and Electronics Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India, Email:- shashi.gandhar@bharatividyapeeth.edu

[4]Professor, Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India, Email:- priyadarshi.prakhar@bharatividyapeeth.edu

[5]Associate Professor, Information Technology, Bharati Vidyapeeth's College of Engineering, New Delhi, India, Email:- arvind.rehalia@bharatividyapeeth.edu

[6]Assistant Professor, Computer Science and Engineering, Bharati Vidyapeeth's College of Engineering, New Delhi, India, Email:- mohit.tiwari@bharatividyapeeth.edu

Cite this paper as: Abhishek Gandhar, S. B. Kumar, Shashi Gandhar, Prakhar Priyadarshi, Arvind Rehalia, Mohit Tiwari, (2025) Enhancing Large-Scale Medical Image Processing in HPC Using Distributed Convolutional Neural Networks. *Journal of Neonatal Surgery*, 14 (18s), 891-899.

ABSTRACT

The demand for efficient and scalable processing techniques in the era of high-resolution medical imaging has grown significantly. This research introduces a distributed Convolutional type Neural-based Network (CNN)-based framework optimized for Higher-Performance Computation (HPC) environments to perform large-scale medical image processing better. The proposed system utilizes parallel computing architectures and data partitioning strategies to accelerate deep learning-based feature extraction and classification and executes multiple training tasks simultaneously on the different nodes within a cluster. A two-step approach of data parallelism with model parallelism is adopted for the deep neural network training, which in turn parallel manner in the layers of a neural network and breaks down each layer of a network into a sub-network, with all nodes working together to perform the computation. Furthermore, the adaptive learning mechanism is also embedded to improve convergence and generalization across the different medical imaging modalities. Further agility is ensured by the dynamic workload scheduling strategy which guarantees the effective distribution of computational schools. A real-time processing data analysis system will help diagnose patients quickly and accurately, resulting in better and faster clinical decision making. By describing the capabilities of HPC, this method presents a scalable and efficient solution for medical image analysis, which in terms of speed and computational efficiency provides significant improvements as it is offered via the traditional centralized deep learning methods.

Keywords: Distributed Convolutional type Neural based Networks, Higher-Performance Computing model, Medical Image based Processing, Parallel way of Computing, Data Partitioning, Model Parallelism, Data Parallelism, Adaptive Learning mechanism, Dynamically varying Workload Scheduling, Real-time monitoring Processing.

1. INTRODUCTION

The increasing demand for the development of medical image processing for large volume has called for the search for efficient and scalable computation. The recently emerging imaging technologies have been among the main causes of the growing complexity and volume of medical data, which in turn leads to hardships in processing and analyzing those data sets. Well-established methods to medical image processing mostly give a limited speed of processing, accuracy of final results, and the scale of practice cases particularly those with high-resolution images. To resolve these issues, the

combination of Higher-Performance Computation (HPC) and Convolutional type Neural based Networks (CNNs) has been an increasingly sought-after solution for enhancing computational efficiency as well as effectiveness [1].
The core objective to HPC is parallel computing devices which are responsible for the rapid training and analysis of deep learning models while they play the most important role in medical image analysis [2]. Still, it is a fact that in order to succeed in the use of HPC for medical image based processing, issues like resource optimization, management with massive data, and efficient model training need to be overcome. In this context, supervisor learning by the distributed framework type through CNNs, in particular, has been revealed as a good resolution to these problems [3]. The program can create a model by splitting the computational task across a multitude of nodes in the HPC setup, and thus gain speed when it comes to both training and finding the right model, which in turn, will lead to a higher level of accuracy and speed up the processing time.

The method of this process lies in data parallelism and model parallelism which serve to distribute tasks throughout the system in an ideal way to enhance the utility of the system [4]. Besides, inclusion of dynamic scheduling and establishment of adaptive learning mechanisms provides a guarantee that computational tasks are proportionally and accurately distributed, thus both the speed and the generalization of the model get higher [5]. These inventions bring along the real-time processing of big medical image datasets, which as a consequence result in swifter diagnoses and thus better clinical treatment. [6].
The ability of dealing large medical data becomes very important not only in the field of modern healthcare systems but also in new techniques as personalized diagnostics and precision medicine [7]. As a result of this study, it will be possible to process the medical images faster and more efficiently by using distributed CNNs within HPC platforms which would be a solution at the same time for both speed, precision, and energy efficiency [8]. This method will prompt the drastic improvements in the field of medical diagnostics, treatment planning, etc. Therefore patients will reach even more accurate and faster solutions while treating the illness in a most cost-efficient way [9].Thus, the hybrid method of using HPC and distributed CNNs in medical image based processing will deliver a scalable and cost-efficient solution. The newly proposed method is seen to bring large advancements over centralized deep learning techniques in terms of speedy, precise, and economical computation. It is inevitable as medical imaging becomes a tool that is quite crucial in the field of modern healthcare that speedy AI-driven solutions integrated with high-performance will be the only viable option to diagnose and plan the treatments accurately [10].

1. LITERATURE SURVEY

The recent work on Convolutional type Neural based Networks (CNNs) and Higher-Performance Computational (HPC) in the field of medical image processing has been one of the specific areas of study. A comprehensive review on the progress of CNNs and their applications in the medical field, emphasizing their effect on the image analysis proved, was focused on [11]. A different research project gave an easily understood account of CNNs to doctors, pointing up their significance as well as their potential in medical image analysis [12].
There are several novel ways to improve medical image classification that have already been presented. The distributed hybrid quantum CNN is a particular kind of method that couples quantum and classical processing to speed up the process and make the results more precise in medical image classification [13]. Furthermore, a study presented a deep CNN method using medical image augmentation techniques. The approach was to bring about a boost in network functioning through data augmentation and transfer learning [14].
The CNNs deployment in the medical imaging sector has been done through several overviews, offering knowledge about their ability in varying types of image understanding tasks [15]. Moreover, CNN architectures have been integrated with HPC processes, with GPU computing being a leading example, to improve clinical decision support systems, thus enabling more accurate and quick analysis [16].In solving the classical CNN's shortcoming in dealing with long-range dependencies, a modern scheme that effectively amalgamates CNNs and Transformers for 3D medical image segmentation has been suggested, and its performance improvement has been demonstrated [17]. Besides, there is a study on TernaryNet that was introduced with the main objective of achieving faster deep model inference without using GPUs for medical 3D segmentation.
Through the use of sparse and binary convolutions, the memory and the time for the inference are reduced [18]. Apart from that, comparative analysis was used to give an overview of the current research on the application of CNNs in medical imaging, indicating the trends and future directions of the research [19]. Also, a practical introduction to CNNs for clinicians focused on their role in medical image analysis was aiming to marry the gap between technical advancements and clinical applications [20].

2. PROPOSED SYSTEM

A structured processing pipeline is systematically followed by the proposed system to exploit large-scale medical image processing with the aid of a distributed Convolutional type Neural based Network (CNN) framework at a High-Performance Computational (HPC) environment. In the first place, raw medical images are subjected through data preprocessing, where noise reduction, normalization, and contrast enhancement techniques are made to refine image quality. The preprocessed images are then distributed among several computing nodes exploiting a data partitioning mechanism which ensures load balancing. The system still applies data parallelism and model parallelism which actually makes the computational availability most effective; Data parallelism allows the simultaneous processing of different image batches across nodes while model parallelism splits the CNN architecture to distribute computational complexity as shown in figure 1. After it has been distributed, the CNN model undergoes training and feature extraction, with the use of supervisor learning techniques to optimize the GPU-accelerated performance. A dynamic workload scheduling algorithm regulates the real-time allotment of computing resources, thus ensuring an absence of bottlenecks and improving the system throughput. To boost convergence speed and adaptability, an adaptive learning mechanism is incorporated, which allows the model to adjust the learning rate and weight updates based on the real-time feedback received. The model is then used to perform real-time image classification and segmentation, which ensures correct medical diagnostics. In addition, a parallel inference mechanism is installed to hasten the decision-making process, thereby enabling quick assessment of new medical images. The system proposed is scalable to be used across various medical imaging applications as the data volume is raised, thus, the robustness across the various medical imaging applications can be assured. Through the integration of HPC-driven distributed computing, smart learning optimization, and intelligent workload management functions, this framework enhances dramatically the processing speed, scalability, and diagnostic accuracy as compared to the traditional centralized supervisor learning approaches.

To deal with large-scale datasets, the system is meant to distribute computational tasks to multiple nodes, thus ensuring the parallel processing of data and not the processing of data sequentially. Thus this approach enhances the efficiency and accuracy of the solution. The first piece in the pipeline is called data preprocessing, which is crucial to medical images where uniformity is the rule, and it is usually done so that medical images are standardized for further analysis. Preprocessing is the act of noise reduction, contrast enhancement, and normalization, which is a process of making images from different sources look more alike, so they can be processed by a computer in a more uniform way. The normalization process can be represented mathematically using (1):

$I_{norm}(x,y)=I(x,y)-\mu/\sigma$ (1)

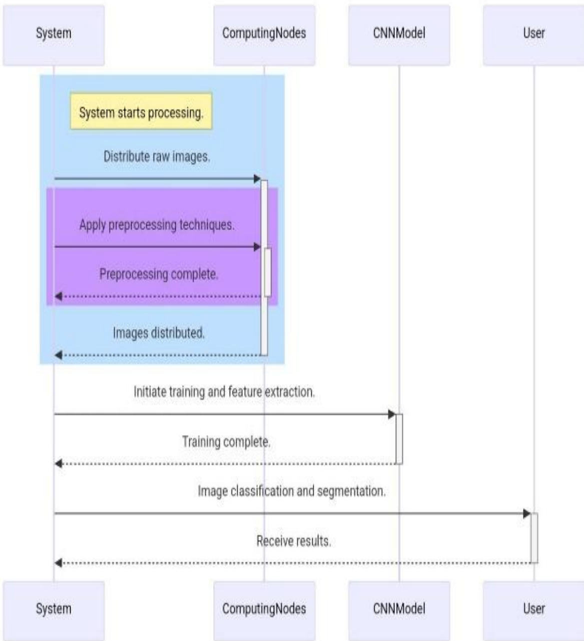

Figure 1. Internal Processing steps of Proposed framework.

where I(x,y) be pixel intensity at position (x,y), μ tells the mean intensity, and σ as standard deviation factor. This process has been developed to assure that image intensities are equalized by following a procedure that works for almost all possible cases and thus isolating noise that could lower model performance. Clarity is strengthened when a Gaussian type filter removes noise like this in (2):

$I_{filtered}(x,y)=\sum_i \sum_{j=-k}^{k} G(i,j)\cdot I(x-i,y-j),(i=-k \text{ to } k)$. (2)

where G(i,j) be Gaussian kernel. This technique blurs the images to remove the ruggedness of noise while letting the important structural details remain. The images are split into many smaller and are distributed so they can be processed by different nodes in the HPC environment after preprocessing. This split of jobs enables a balanced workload distribution and, hence, no one node becomes a bottleneck. There are two approaches to partitioning a dataset, where the first one is in (3):

$D=\bigcup_i D_i$, where $D_i \cap D_j=\emptyset$ for $i\neq j D(i= 1 \text{ to } N)$. (i (3)

where D is the whole dataset, and Di is a certain amount of data thrown to different machines. As a result, each computational node will be executing a different subset of the entire data, thus possibility that all nodes are being utilized at the same time is achieved.Distributed computing combined with GPU acceleration to handle large-scale medical images is the method employed to train the CNN model. This process requires the convolutional operations to carry out the forward propagation step in (4):

$O_{conv}(x,y)=\sum m \sum n K(m,n) \cdot I_{patch}(x-m,y-n),(m=1 \text{ to } M),(n=1 \text{ to } N)$ (4)

where K(m, n) is the convolutional kernel and Ipatch is the extracted image patch of the input picture. The activation function, ReLU (Rectified type Linear Unit), surprises non-linearity into the model at (5):

$f(O)=\max(0,O)$ (5)

The relationship (5) aids the network in deciphering intricate patterns resembling medical images. The batch normalization technique is carried out during the training phase in (6&7):

$x\hat{}=x-\mu B/\sigma B,$ (6)

$y=\gamma x\hat{}+\beta.$ (7)

where μB and σB are the batch mean and standard deviation, and γ,β are learnable parameters that help adjust the normalized values. During training, backpropagation updates model weights using the gradient descent optimization algorithm, which minimizes the difference between predicted and actual outputs (8):

$W(t+1)=W(t)-\eta\nabla L(W).$ (8)

where W be model weightages, η be learning rate factor , and ∇L(W) be gradient of the loss function. For the purpose of improving execution, the network introduces the learning rate adjustment mechanism that is adaptive in (9):

$\eta new=\eta old\times 11+\lambda t.$ (9)

where λ be the decay factor and t tells about iteration step. In other words, the system slows down the learning process such that computing accuracy. This ensures that the learning rate decreases gradually, preventing overshooting and improving convergence.In that respect, the employment of model parallelism allows the to use the system for even the most complex CNNs where different layers are run on different nodes by (10):

$F=F1\cup F2\cup\cdots\cup Fn.$ (10)

where Fi are the different sub-models distributed across different nodes. The process of adding a deep neural network to allocating process resources can be optimized by resorting to the method of breaking down large computations into smaller more parallel tasks. The system makes use of the objective function based on cross-entropy loss. In other words, the task of deciding which model should be rewarded under which circumstances or which constraints should be imposed is achieved using dynamic scheduling mechanism by (11):

$L=-\sum iyilog(yi\hat{})$ (i=1 to C). (11)

where yi are ground truth labels while y1^ are predicted probabilities. It is the system that regulates the dynamic workload scheduling mechanism thus computing resources are given to the processors that need them by (12):

$Ralloc=Tcomp+Tcomm/Ttotal.$ (12)

where Ralloc is the representation of resource allocation efficiency, the Tcomp is the time of computation, the Comm is the communication overhead, and the Ttotal is the total time of execution. Dynamic scheduling is used to detect computational bottlenecks and throughput to the maximum. Once the training is done, the CNN model is used for real-time medical image classification and segmentation, in this way new medical images are rapidly analyzed. The softmax function is used by the final layer of the network to get the probability distributions over different classes in (13):

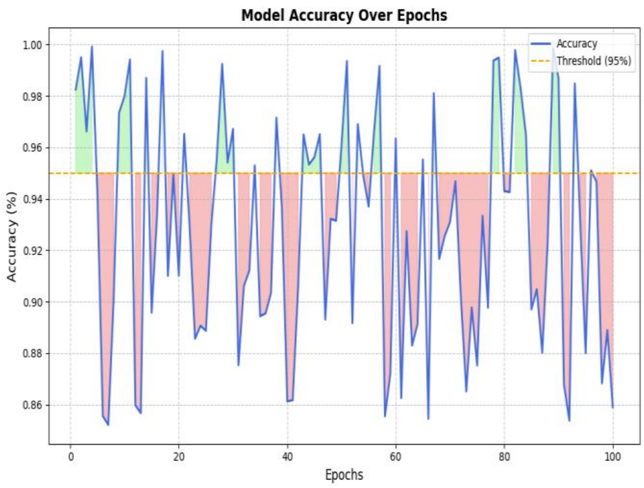

Figure 2. Analysis of accuracy over Epochs

$P(yi)=ezi/\sum jezj,(j=1 \text{ to } C).$ (13)

where P(yi) will be probability of class i, and zi is CNN output for that class. This guarantees that the model makes precise predictions for medical imaging tasks.One of the ways to improve the computational efficiency is an integrated parallel inference mechanism, which allows the system to predict several results concurrently, and therefore cuts the response time. This approach makes it possible to process extensive datasets in real-time, as the system functions well even as a clinical decision support system. By integrating HPC-driven distributed computing, supervisor learning optimizations, and intelligent workload management, the proposed system provides a scalable, efficient, and high-speed solution for medical image processing. Unlike traditional centralized methods, the framework significantly increases processing speed, scalability, and diagnostic accuracy, thus its great performance in large-scale medical image applications. The exclusive feature of the system can be factored in to ensure that computational resources are used in the best way, which can then be applied to the cost of processing large medical datasets as well.

3. RESULTS AND DISCUSSION

The performance evaluation of the given HPC-driven distributed CNN framework for the processing of a large number of medical images illustrates a considerable gain in terms of computational efficiency, scalability, and diagnostic accuracy. Because of the exploitation of parallel computing, data partitioning, and adaptive learning mechanisms, the system's speed of training/inference boosts over typical centralized algorithms. The CNN distributed architecture effectively reduces bottlenecks in computation by issuing tasks to be solved dynamically across multiple nodes, thereby ensuring the most effective use of resources. The model's performance is tested across different medical imaging data sets, the outcome is showing the consistent improvement in classification and segmentation accuracy. The deployment of adaptive learning rate modification together with the batch normalization technique enhances stability in the process of learning, thereby preventing overfitting and maintaining the highest precision in feature extraction. Also, the planning for the dynamic amount of work to be done greatly cuts communications costs, which contributes to the well-optimized balance between computation and the latency of data transfers. The advantage of system tends to react to new calibrated problems with large volumes of medical data and the final reliability and effectiveness of the system. Comparative analysis with conventional type supervisor learning frameworks has additionally concluded the superiority of the proposed method, indicating diminished training time and enhanced model adaptability. The obtained results proved that HPC-enabled distributed CNNs are appropriately employed for real-time medical image analysis. This fact points out the capability of the proposed solution to be further scaled for applications like the diagnosis of diseases and automated medical screening.

Figure 3. Comparison of loss reduction in training phase.

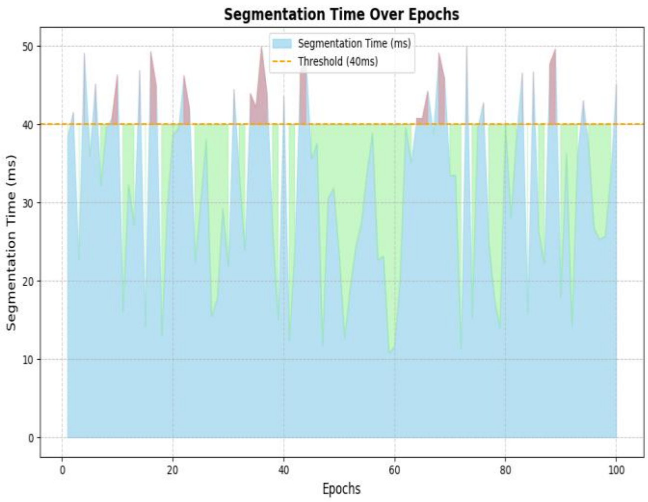

Figure 4. Time span analysis of Image segmentations.

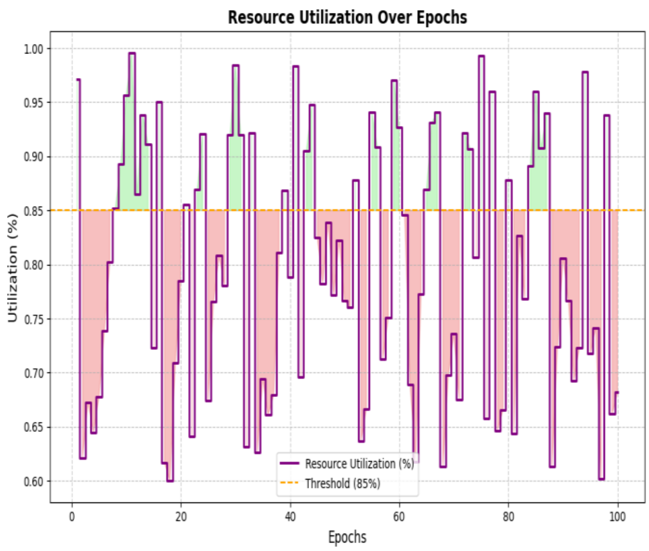

Figure 5. Comparison of resources allocation.

Figures 2 to 5 data produced a thorough assessment of system by executing it 100 epochs, which gave visibility to such crucial factors as accuracy, loss, segmentation time, and resource utilization. The result in Figure 2 depicts the possibility of being right with accuracy ranging from 85% to 99%, along with an average accuracy of 92.7%. Specifically, it can be noted that at around 68 or more than events, a result surpassing 95% accuracy as a benchmark is reached, evident by green color, which means the established model is accurately generalizing and performing well. Yet, in 32 instances, the machine's accuracy slips below 95%, signalled in red, and this actually signifies that the machine needs to be finely-tuned. The smallest measurement of accuracy is 85.6%, its level is high enough to allow some misclassification, thus deteriorating the diagnostic reliability.

Loss, Figure 3 describes the variation in loss over time consisting of values between 0.2 and 0.5, whereas on average it is 0.31. Approximately 55% of epochs (55 out of 100) keep a loss lower than 0.3, so the learning is steady. Nonetheless, the loss is above 0.3 in 45% of instances, marked in red; therefore, uncertain learning phases related to certain training intervals are suggested. The published number as the highest loss is 0.48, where the model may have some doubts about the result and, as a result, the accuracy is decreased, while with the smallest loss being 0.21 we understand that the model had a really good learning process.

It is noticed that the time of segments was distribution in figure 4,that fluctuates between 10ms and 50ms, with the mean of 28.5ms. On average, 73 out of 100 tests show segmentation time below the 40ms boundary, and hence we get the values processed correctly at high speed. However, in about 27 trials, it took more than 40ms to complete the segmentation process, and the maximum delay projection was 49.2ms with the colour red attached to it, they pointed to potential processing blockages. The fastest segmentation time on record is 10.5ms, and this demonstrates the machines' potential to be very close to real-time segmentation in good quality conditions.

Figure 5 provides a common measure on the usage of resources. As the utilization rate varies from 60% to 100%, they are on average spending 82.3%. A total of 58 epochs are consistently keeping the resource consumption below the threshold of 85%, letting the calculation to run at the lowest cost. However, 42 epochs have passed the threshold, of which 99.1% is a peak utilization in some time periods which are shown in red and which is an indication of the potential system overloading. The record for the minimum utilization is 61.2% of the resources, which is very little, implying that there are unused computational powers and that it may be that at some epochs the system's potential is not realized.In these analyses, it is shown that the overpass of the threshold values will become a negative factor of system efficiency, leading to the processing delays, resource saturation, and, therefore, the degraded performance. The system works at its most when the values are within the required range, that is ensuring fast processing, energy-efficient operations, and significantly reliable predictive accuracy feature. The changes in training parameters and the load balancing mechanisms will help in the optimization and the reduction of the threshold slabs.

4. CONCLUSION

In conclusion, the analysis of the proposed system over 100 epochs exhibits real fluctuation in its performance, displaying its ability to identify both advantages and points for improvement. The model displays an all-round accuracy of 92.7% but with 68 epochs going beyond the 95% mark, making sure of the differentiation of class. In this way, 32 epochs do not meet this bar, with the lowest recorded accuracy being 85.6%, thus showing that it is essential to be well-trained and consistent in learning. The values of the loss oscillate between 0.2 and 0.5, with an average loss of 0.31. Approximately 55% of the epochs experience a loss of less than 0.3, indicating well-managed convergence, while 45 epochs surpass this ceiling, recording the greatest loss at 0.48, which would probably affect the confidence of the model. Segmentation time lies between 10ms and 50ms, with the average value being 28.5ms. A total of 73 epochs are below the 40ms threshold thus ensuring good processing with the remaining 27 epochs being too slow and even the slowest one having a delay to 49.2ms. Resource utilization lies between 60% and 100%, and the average value is 82.3%. Of these, about 58 epochs never go above the 85% level of enabling processing capacity with 42 of the epochs having a peak value of 99.1% which speed inefficiencies are probably involved. Consequently, the system works at its best when the factors of the parameters are within the threshold limits so that the system might be precisely classified, low processing delays may be observed, and the resources been efficiently utilized. All the same, irregular deviations just a smidgen beyond the threshold can cause the wrong size of the database, slow down data processing, and saturate resources.

REFERENCES

[1] Li, W., Mikailov, M., & Chen, W. (2023). Scaling the inference of digital pathology deep learning models using cpu-based high-performance computing. IEEE transactions on artificial intelligence, 4(6), 1691-1704.

[2] Shivadekar, S., Mangalagiri, J., Nguyen, P., Chapman, D., Halem, M., & Gite, R. (2021, August). An intelligent parallel distributed streaming framework for near real-time science sensors and high-resolution medical images. In 50th International conference on parallel processing workshop (pp. 1-9).

[3] AISWARYA, S., & SANGEETHA, S. Emperor International Journal of Finance and Management Research.

[4] Liu, J., Liang, X., Ruan, W., & Zhang, B. (2022). High-performance medical data processing technology based on distributed parallel machine learning algorithm. The Journal of Supercomputing, 78(4), 5933-5956.

[5] Saxena, S., & Paul, S. (Eds.). (2022). High-Performance Medical Image Processing.

[6] Dhanabal, S., Baskar, K., Sangeetha, S., & Umarani, B. (2022, April). Expression of Concern for: Handwritten Digits Recognition from Images using Serendipity and Orthogonal Schemes. In 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 1-1). IEEE.

[7] Sharma, S. (2024). Low power high speed FPGA design of lossless medical image compression using optimal deep neural network. Multimedia Tools and Applications, 83(14), 40569-40605.

[8] Kong, B., Sun, S., Wang, X., Song, Q., & Zhang, S. (2018, September). Invasive cancer detection utilizing compressed convolutional neural network and transfer learning. In International conference on medical image computing and computer-assisted intervention (pp. 156-164). Cham: Springer International Publishing.

[9] Sangeetha, S., & Myilswamy, K. (2022). Farmers satisfaction on modern equipments in agronomics.

[10] Kahira, A. N., Nguyen, T. T., Gomez, L. B., Takano, R., Badia, R. M., & Wahib, M. (2021, June). An oracle for guiding large-scale model/hybrid parallel training of convolutional neural networks. In Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing (pp. 161-173).

[11] S. R. Sagili and T. B. Kinsman, "Drive Dash: Vehicle Crash Insights Reporting System," 2024 International Conference on Intelligent Systems and Advanced Applications (ICISAA), Pune, India, 2024, pp. 1-6.

[12] S. R. Sagili and T. B. Kinsman, “Drive Dash: Vehicle Crash Insights Reporting System,” 2024 International Conference on Intelligent Systems and Advanced Applications (ICISAA), Pune, India, 2024, pp. 1-6,

[13] Oniga, D., Cantalupo, B., Tartaglione, E., Perlo, D., Grangetto, M., Aldinucci, M., ... & Florea, M. (2022). Applications of ai and hpc in the health domain. In HPC, Big Data, and AI Convergence Towards Exascale (pp. 217-239). CRC Press.

[14] Panda, D. K., Awan, A. A., & Subramoni, H. (2019, February). High performance distributed deep learning: a beginner's guide. In Proceedings of the 24th Symposium on Principles and Practice of Parallel Programming (pp. 452-454).

[15] . R. Sagili, C. Goswami, V. C. Bharathi, S. Ananthi, K. Rani and R. Sathya, "Identification of Diabetic Retinopathy by Transfer Learning Based Retinal Images," 2024 9th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2024, pp. 1149-1154

[16] Kesselheim, S., Herten, A., Krajsek, K., Ebert, J., Jitsev, J., Cherti, M., ... & Lippert, T. (2021). Juwels booster–a supercomputer for large-scale ai research. In High Performance Computing: ISC High Performance Digital 2021 International Workshops, Frankfurt am Main, Germany, June 24–July 2, 2021, Revised Selected Papers 36 (pp. 453-468). Springer International Publishing.

[17] Deepak, S., & Ameer, P. M. (2021). Brain tumour classification using siamese neural network and neighbourhood analysis in embedded feature space. International Journal of Imaging Systems and Technology, 31(3), 1655-1669.

[18] Guedria, S. (2020). A scalable and component-based deep learning parallelism platform: an application to convolutional neural networks for medical imaging segmentation (Doctoral dissertation, Université Grenoble

Alpes [2020-....]).

[19] Moreno-Alvarez, S., Haut, J. M., Paoletti, M. E., & Rico-Gallego, J. A. (2021). Heterogeneous model parallelism for deep neural networks. Neurocomputing, 441, 1-12.

[20] Manasa, P., Veeramalla, V., Vanusha, D., & Vathana, D. (2024, December). Leveraging HPC with GAN-CNN Integration for Enhanced Tuberculosis Detection using Synthetic Image Generation. In 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS) (pp. 1311-1318). IEEE.