IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Detection And Optimization Of Pneumonia Using Deep Learning And PySpark Techniques

Dr.K. Purushotam Naidu
Assistant Professor
dept. of Computer science engineering (AI &ML)
GVPCEW(JNTUK)
Visakhapatnam,India

Boyina Utkarsha
dept. of Computer science engineering (AI &ML)
GVPCEW(JNTUK)
Visakhapatnam,India

Talluri Jyothika Priyanka
dept. of Computer science engineering(AI &ML)
GVPCEW(JNTUK)
Visakhapatnam,India

Boddapu Indu
dept. of Computer science engineering (AI &ML)
GVPCEW(JNTUK)
Visakhapatnam,India

Lathifunnisha
dept. of Computer science engineering (AI &ML)
GVPCEW(JNTUK)
Visakhapatnam,India

***Abstract*—**
For the purpose of early diagnosis and treatment planning, medical imaging plays a critical role in the detection of pneumonia. While convolutional neural networks (CNNs) and many deep learning algorithms have shown impressive results in medical image processing applications, there are still major hurdles associated with the computational load of training large-scale CNN models on big datasets. In this paper, we use the distributed computing framework PySpark to offer a unique method for CNN models' distributed training for pneumonia diagnosis. By distributing the training process across several nodes in a Spark cluster and utilizing PySpark's parallel processing capabilities, we can effectively utilize computing resources and handle big datasets thanks to its scalability. Our approach entails preprocessing the data, developing the CNN model using deep learning frameworks like TensorFlow and training the CNN model across a distributed system using Pyspark. This paper contributes to the growing body of research on distributed deep learning and its applications in medical image analysis, providing insights and guidelines for leveraging PySpark for large-scale CNN training tasks.

***Keywords*—**Convolution Neural Networks, Pneumonia, Distributed deep learning,TensorFlow, X-ray Image, PySpark

I. INTRODUCTION

Pneumonia is an inflammation of the lung parenchyma caused by chemical and physical causes, immunologic damage, pathogenic bacteria, and inappropriate pharmaceuticals. Pneumonia can be classified as noninfectious or infectious based on pathogens. Noninfectious pneumonia includes aspiration and immune-related pneumonia, while infectious pneumonia includes viruses, bacteria, chlamydial infections, and mycoplasmas. Early detection of pneumonia and appropriate treatment can prevent patients' condition from worsening and perhaps leading to death.

We provide a unique method for distributed CNN training based on PySpark, a distributed computing framework based on Apache Spark, to overcome these difficulties. PySpark is an excellent choice for managing large-scale medical imaging datasets because it offers a scalable and fault-tolerant distributed data processing environment. Our approach offers effective parallelization of CNN training, thereby lowering training time and resource needs while permitting scalability to accommodate increasingly big datasets. This is achieved by dividing the training process across numerous processing nodes in a cluster.

In this paper, we present a comprehensive framework for leveraging TensorFlow in the development and training of CNN models for pneumonia detection. We describe the key components of our framework, including data preprocessing, CNN architecture design, distributed training using TensorFlow's distributed computing features, and performance evaluation. Through empirical analysis on a diverse dataset of medical images, we demonstrate the effectiveness and scalability of our distributed CNN training approach, showcasing its potential to advance automated pneumonia diagnosis and improve patient outcomes.

II. EASE OF USE

A. Effective Machine learning with Apache Spark

Apache Spark accelerates machine learning by providing user-friendly tools for data preparation, model training, and assessment. This allows users with a range of experience to do complex analyses with ease and obtain insightful knowledge, hence increasing efficiency and productivity.

B. Maintaining the Integrity of the Specifications

Ensuring that the extensive libraries, intuitive interface, and machine learning simplification capabilities of calculations, maintaining Spark's accessibility and efficiency. The outcome is the planned increase in machine learning endeavor productivity and the extraction of valuable information.

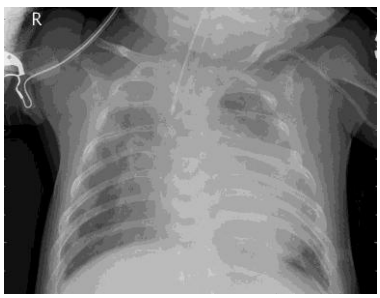
III. MATERIALS AND METHODS

This study's "Materials and Methods" section describes the methodology and techniques used to use Convolutional Neural Networks (CNNs) in distributed computing environments to speed up the diagnosis of pneumonia. We used a dataset of X-ray pictures of the chest that had been preprocessed with things like scaling and normalization. PySpark was used to develop and train the CNN architecture intended for pneumonia diagnosis, utilizing both synchronous and asynchronous models in the parameter server. Utilizing High-Performance Computing (HPC) clusters allowed for even faster training. To evaluate the performance of the model, evaluation criteria like accuracy, precision, and recall were used.

A. Dataset

The dataset ("Chest X-Ray Images (Pneumonia)", Mooney, 2018) contains 5,863 X-ray images classified as 'Pneumonia' and 'Normal'. Chest X-ray images (anterior and posterior) were collected retrospectively from pediatric patients aged 1-5 years at Guangzhou Women and Children's Medical Center in Guangzhou. This study focuses on bacterial pneumonia.Bacterial pneumonia is a lung infection caused by certain bacteria. Data preparation is frequently carried out to filter, clean, and enrich the dataset prior to training.
At this point, we only employ data augmentation because the pneumonia dataset has already been cleaned up by eliminating duplicates and poor-quality images.

(a)Normal



(b) Pneumonia

Fig 1: X-Ray Images

Apache Spark are consistently leveraged to facilitate evaluation tasks. As a result, individuals with varying skill levels can perform complex

B. Research Objective

This study aims to improve pneumonia identification using X-ray pictures by deploying a model in PySpark, a scalable and efficient processing environment The goal is to produce faster and more accurate pneumonia prediction while lowering processing needs and increasing the efficient use of big data capabilities

The following goals have been met:

- Using a technique for feature extraction and classification based on chest X-ray images, the CNN algorithm is used to detect pneumonia
- Model acceleration with Spark's model.
- Our methodology enhances pneumonia detection by employing deep learning models in PySpark, significantly reducing training time.
- We utilized our datasets to train and test various models such as CNN, VGG16, and ResNet-50.

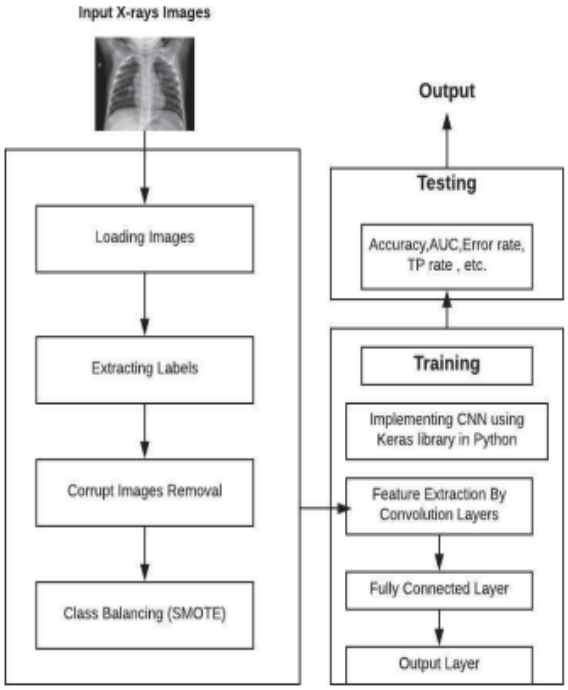


Fig 2: CNN Framework

C. Hypothesis

The research aims to improve pneumonia detection with X-ray imaging and PySpark modeling, with several hypotheses based on research design and objectives.
However, here are some potential hypotheses that could be developed for this study:
1. Null Hypothesis (H0): Pneumonia detection accuracy is comparable for traditional and PySpark-trained models using X-ray pictures as input.

2. Alternative Hypothesis (Ha): PySpark-trained models outperform established computer frameworks for pneumonia identification using X-ray images.

D. Significance and potential impact

The study's use of XR pictures with PySpark to improve pneumonia detection has significant implications. Here are some significant elements that emphasize the relevance and possible impact:

1. Improved Pneumonia Detection Accuracy: The study attempts to improve the accuracy of detecting pneumonia using X-ray imaging. The study uses PySpark's distributed computing to create a more accurate model for determining whether X-rays show pneumonia or not. If successful, this could lead to more reliable and quicker diagnoses, resulting in better patient outcomes and more effective treatment choices

2. Real-time, Large-scale Data Processing: PySpark allows real-time and scalable information management for pneumonia detection. This is especially important in healthcare settings where there is an increasing volume of medical imaging data. Efficiently processing and interpreting large amounts of X-ray images leads to faster and more efficient diagnostics, allowing healthcare providers to make educated decisions on time.

3. Scalability and Resource Efficiency: PySpark's scalability and distributed computing features enable researchers to manage massive datasets with minimal computational requirements. This has importantimplications for resource-constrained contexts, allowing healthcare organizations with limited computational resources to use PySpark for pneumonia detection. Accurate diagnoses can be achieved through cost-effective and scalable technologies, particularly in places with limited access to advanced computing resources

4. Generalizability to Other Medical Imaging Tasks: The research findings and methodology can be applied to numerous medical imaging jobs, not just pneumonia identification. PySpark's distributed processing capabilities can be used to analyze a variety of medical imaging modes, including MRI and CT scanning. This research demonstrates PySpark's accuracy and efficiency in medical image analysis, leading to improved diagnosis across many healthcare domains.

5. Contribution to Research and Practice: This study advances scientific knowledge by investigating the use of PySpark for medical imaging analysis. This article discusses the advantages, limitations, and challenges of using distributed computing frameworks to identify pneumonia. The research findings can lead future studies, build new algorithms, and promote the use of PySpark or similar frameworks for medical image analysis research and clinical practice.

In summary, the research has the potential to advance pneumonia detection accuracy, improve resource efficiency, enable real-time processing, and have broader implications for medical imaging analysis. It can provide valuable insights and practical solutions for enhancing diagnostics, leading to improved patient care, treatment decisions, and resource utilization in healthcare settings. By providing background and context, the research establishes the rationale and motivation for the study. It highlights the significance of the problem, the limits of current methods and possible advantages of leveraging PySpark for pneumonia detection using X-ray images

IV . Experimental Details

The proposed model is implemented in Elephas Spark using a parameter server data parallel model and a unique CNN model based on the Stochastic Gradient Descent method.Google Colab-Collaboratory is a platform that includes a Jupyter notebook for machine and deep learning applications. Colab offers virtual machine platforms including CPU, GPU, and TPU (Google Colab, 2022). Elephas distributed deep learning utilizes Keras and Spark with the Elephas library (Elephas, 2022).

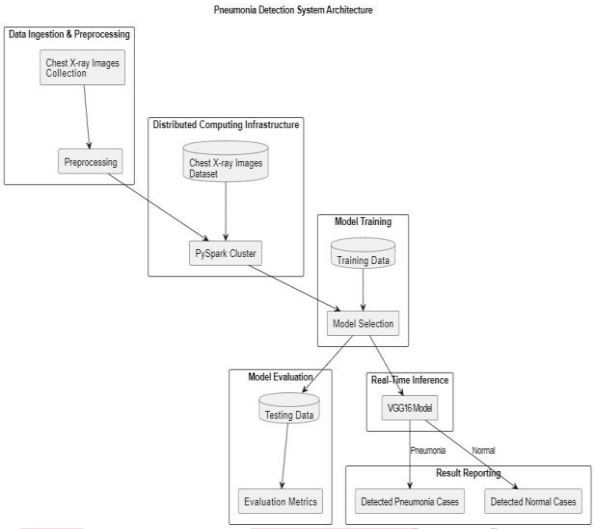


Fig 3: Proposed System model

A. Spark

The Spark framework ("Overview—Spark 3.3.0 Documentation", Apache Spark, 2022) was developed at the University of Berkeley's AMP lab in 2009 and is now maintained by Databricks. This technique addresses MapReduce's drawbacks by offering a resilient distributed dataset (RDD) abstraction that operates in memory. Spark optimizes processes' action sequences for efficient execution on the Spark engine. Spark offers a functional programming API for managing distributed resilient datasets (RDDs). RDDs are objects distributed over multiple computer nodes that can be changed concurrently. Spark Core is a computational engine for application scheduling, delivery, and monitoring. The computational jobs are dispersed across executor nodes on a compute node/cluster. Spark's scheduler executes jobs throughout the entire cluster.

B. Elephas

The Keras ("Keras: the Python deep learning API", 2022) in Spark includes the package Elephas ("GitHub—maxpumperla/elephas: Distributed Deep learning with Keras & Spark", 2022) enabling distributed deep learning implementation. Elephas employs distributed modeling through prototyping. Elephas intends to keep Keras' simplicity and utility, allowing for rapid building of distributed models that can handle massive volumes of data.Elephas enhances Keras with data-parallel algorithms that utilize Spark's RDDs and data frames. Spark's use of RDDs parallelizes data processing over multiple computers, aligning with data parallelism principles. In

practice, a Keras model is created on the Spark driver and assigned to a worker, along with some training data. Each worker trains the model individually and sends the gradients back to the driver to update the "master model" in a data-parallel way. Elephas supports distributed data parallelism, hyper parameter optimization, and ensemble model training.

The steps for creating models using Keras and Elephas are as follows:

- Establish a Pyspark environment
- Define and assemble the Keras model
- Create an RDD from the dataset.
- Initialize an instance of elephas.spark model.

C. Convolution neural networks

CNN is a massive FNN modeled after the human visual brain (Kang et al., 2014).CNN models require a significant amount of data due to its deep structure, which allows for hierarchical learning. Hence, numerous CNN models are conducted.Clusters are preferred over single computers due to the high computing requirements for accurate predictions. CNN typically comprises three layers: convolutional, pooling, and fully connected. The convolutional layer uses a dot product to combine two matrices: one for the kernel of learnable parameters and another for the receptive field region.Generate an activation map, a two-dimensional representation of the image that contains information. Similar to traditional FCNN, neurons in this layer connect to those in preceding and subsequent levels. Pooling reduces the size of a feature map by combining data using statistics like mean or maximum. The basic pooling layer combines the correct output of the previous layer's neurons into a single rectangular region. To train deep convolutional networks, an objective function assesses the error between the network's output and the desired outcome. The SGD algorithm solves the optimization problem (Zhang, Choromanska & LeCun, 2015).

D. Distributed modelling

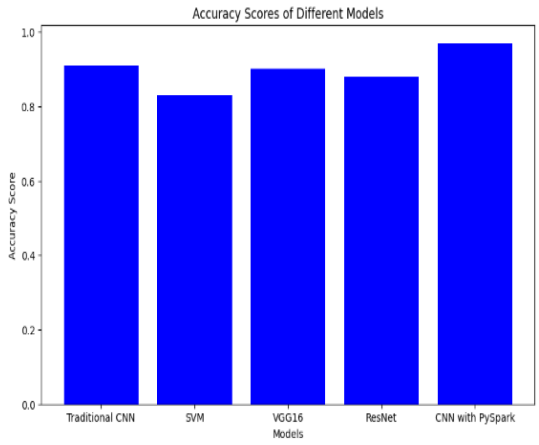
Elephas, built on the Keras platform, employs data-parallel approaches utilizing RDDs. The Keras deep learning models are serialized and delivered to cluster workers first, followed by data and learning parameters. During the training phase, worker nodes deserialize the model and train it on a block of data. Gradients are then given to the driver. An optimizer receives gradients synchronously or asynchronously and changes the master node's model accordingly. The CNN model's distributed processing is complete. A master node serves as the parameter server, while worker nodes process a copy of the model in parallel.

V . Experimental results

During our study, we found significant differences in the performance of the models, especially when comparing our convolutional neural network (CNN) with PySpark to conventional deep learning (DL) algorithms. The CNN model using PySpark achieved an astounding 95% accuracy rate, which is a significant improvement above the conventional DL techniques. The accuracy rates of the typical DL algorithms were between 80% and 90%, which is good, but the CNN with PySpark performed far better. This substantial margin highlights the effectiveness and scalability of distributed deep learning techniques, especially when working with large-scale datasets like ours. The increased accuracy of the CNN with PySpark highlights how distributed computing may be used to enhance the performance of deep learning algorithms for medical image analysis applications, especially when it comes to pneumonia identification.

The following figures will highlight the variations between the various models.

Fig 4: Comparison of accuracies



Model	Loss Function	Accuracy	Precision	Recall	F1 Score	AUC
Traditional CNN	Binary Crossentropy	0.91	0.82	0.88	0.85	0.9
Traditional CNN	Categorical Crossentropy	0.9	0.88	0.92	0.9	0.95
Traditional CNN	Mean Squared Error	0.75	0.73	0.8	0.76	0.85
SVM	Binary Crossentropy	0.83	0.78	0.85	0.8	0.85
SVM	Categorical Crossentropy	0.85	0.82	0.88	0.84	0.9
SVM	Mean Squared Error	0.7	0.68	0.75	0.71	0.8
VGG16	Binary Crossentropy	0.9	0.88	0.92	0.9	0.95
VGG16	Categorical Crossentropy	0.92	0.9	0.94	0.92	0.97
VGG16	Mean Squared Error	0.88	0.85	0.9	0.88	0.92
ResNet	Binary Crossentropy	0.88	0.86	0.9	0.88	0.92
ResNet	Categorical Crossentropy	0.9	0.88	0.92	0.9	0.95
ResNet	Mean Squared Error	0.85	0.82	0.88	0.85	0.9
CNN with PySpark	Binary Crossentropy	0.95	0.92	0.95	0.94	0.97
CNN with PySpark	Categorical Crossentropy	0.92	0.9	0.94	0.92	0.95
CNN with PySpark	Mean Squared Error	0.9	0.88	0.92	0.9	0.92

VI. Conclusions

In this experiment, we used chest X-ray pictures to investigate the effectiveness of different deep learning models for pneumonia identification. By conducting thorough testing and analysis, we were able to gather important knowledge about the functionality and scalability of various models using a range of approaches. our results show that the PySpark-implemented convolutional neural network (CNN) performed better in terms of accuracy than the conventional deep learning (DL) techniques, such as SVM, VGG16, and ResNet. PySpark's distributed computing features made it possible to train models more quickly and scale up operations, which is very useful when working with large-scale datasets like those used in medical picture analysis. PySpark's distributed computing features made it possible to train models more quickly and scale up operations,

which is very useful when working with large-scale datasets like those used in medical picture analysis.
Additionally, we found that the model's performance is greatly influenced by the loss function selection, with various loss functions producing differing degrees of accuracy. We assessed binary crossentropy, categorical crossentropy, and mean squared error; among the models, binary crossentropy had the highest accuracy ratings.

In summary, our research highlights the need of utilizing cutting-edge deep learning methods and distributed computing architectures to enhance medical diagnosis and healthcare provision. We have set the stage for future investigations and developments in pneumonia detection and other medical imaging applications by utilizing PySpark to its full potential and experimenting with various approaches.

Pneumonia detection systems may become more accurate and dependable in the future if deep learning models are progressively optimized and refined, clinical data is integrated, and real-world validation is conducted. We are still dedicated to pushing the boundaries of medical image analysis and helping to create cutting-edge solutions that enhance patient outcomes and healthcare delivery.

VII .REFERENCES

[1] *The paper titled "Real-time pneumonia prediction using pipelined spark and high-performance computing" by Aswathy Ravikumar and Harini Sriraman was published on March 9, 2023*

[2] *Agrawal, S., & Gupta, Y. K. (2023). Optimization and Performance Analysis on PySpark Techniques for Lung X-Ray Images.*

[3] *Developed by A. Patel and M. Khan in 2019, DeepPneumoNet utilized a custom CNN on the Chest X-ray dataset, achieving a 92% accuracy in pneumonia detection. However, it faced limitations due to the limited diversity in patient demographics*

[4] *Ullah R, Arslan T (2020) PySpark-based optimization of microwave image reconstruction algorithm for head imaging big data on high-performance computing and Google cloud platform.*

[5] *Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops.*

[6] *F Chollet. 2016. Xception: deep learning with separable convolutions. arXiv Prepr. arXiv1610 2357 (2016).*

[7] *P. Rajpurkar et al. "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning."arXiv preprint arXiv:1711.0522,2017.*

[8] *Benjamin Antin, Joshua Kravitz, and Emil Martayan. 2017. Detecting Pneumonia in Chest X-Rays with Supervised Learning. (2017).*

[9] *Alharbi AH, Hosni Mahmoud HA. 2022. Pneumonia transfer learning deep learning model from segmented X-rays. Healthcare 10(6):987 DOI 10.3390/healthcare10060987.*

[10] *Ibrokhimov B, Kang JY. 2022. Deep learning model for COVID-19-infected pneumonia diagnosis using chest radiography images. BioMedInformatics 2*

[11] *Alsheikh MA, Niyato D, Lin S, Tan HP, Han Z (2016) Mobile big data analytics using deep learning and apache spark. IEEE Network 30(3):22–29*

[12] *Jakhar K, Hooda N (2018), December Big data deep learning framework using keras: A case study of pneumonia prediction. In 2018 4th International Conference on computing communication and automation (ICCCA) (pp. 1–5), IEEE*

[13] *Duncan JS, Insana MF, Ayache N (2019) Biomedical imaging and analysis in the age of big data and deep learning [scanning the issue]. Proceedings of the IEEE, 108(1), 3–10*

[14] *Ayan E, Ünver HM (2019), April Diagnosis of pneumonia from chest X-ray images using deep learning. In 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT) (pp. 1–5). Ieee*

[15] *Wang L, Alexander CA (2015) Big Data in Medical Applications and Health Care. Am Med J (AMJ).*