Enhanced Medical Image Analysis: Leveraging CUDA for Fast and Accurate Pneumonia Detection with optimized CNNs

by

Md.Waseq Alauddin Alvi
20101153

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

Department of Computer Science and Engineering
Brac University
May 2024

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md.Waseq Alauddin Alvi
20101153

Approval

The thesis/project titled "Enhanced Medical Image Analysis: Leveraging CUDA for Fast and Accurate Pneumonia Detection with optimized CNNs." submitted by

1. Md.Waseq Alauddin Alvi (20101153)

Of Spring, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 2024.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Ashraful Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi
Associate Professor; Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Pneumonia, a known leading child killer and a general health burden, continues to be a major concern due to its high morbidity and mortality rates in the developing world, which calls for prompt and accurate diagnosis. This paper aims at proposing a novel medical image analysis framework that can be used in the enhancement of pneumonia from Chest X-ray images in terms of speed and accuracy. Building on the capability of the Convolutional Neural Networks (CNNs) that have been tuned using NVIDIA CUDA, this strategy enhances the computational capabilities and enables real time analysis. Hence, it meant that we were training a novel deep learning model which was fit for the specific task we were undertaking involving identification of bacterial, viral pneumonia in addition to normal cases. The model finds feature extraction and considers incorporation of advanced layers and/or architectures. By paralleling the codes with Cuda we were able to reduce the time it takes to train and make prediction on models while at the same time not being compromising on the quality of the models. In addition, Our experimental results show that, our CUDA-optimized CNN outperforms and achieve equal or higher accuracy against the traditional methods, all this in a drastically shorter time. There is potential for deploying associated high-resolution diagnostic equipment in clinical environemnt, specifically in situation where decisions are needed quickly. Our self-contrary contributions signify the effectiveness as well as effectiveness of deep learning and high-performance computing to augment the medical diagnostic technique and would open the area to extensive applications of medical image analysis in the future.

Keywords: Pneumonia detection, medical image analysis, convolutional neural networks (CNNs), NVIDIA CUDA, chest X-ray, real-time analysis, bacterial pneumonia, viral pneumonia, high-performance computing, healthcare diagnostics, deep learning

Dedication

With deep gratitude to my supervisor, Dr. Md. Ashraful Alam, whose knowledge and encouragement shaped this work, and to my parents, whose love and sacrifices made it possible.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to my Supervisor Md. Ashraful Alam sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer, we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$Adam$ Adaptive Moment Estimation

$BatchNorm$ Batch Normalization

$CategoricalCross - Entropy$ A Loss Function for Multi-Class Classification

CNN Convolutional Neural Network

$CUDA$ Compute Unified Device Architecture

GPU Graphics Processing Unit

MSE Mean Square Error

$ReLU$ Rectified Linear Unit

SGD Stochastic Gradient Descent

SNN Siamese Neural Network

$Softmax$ A Function that Converts Logits to Probabilities

$SSIM$ Similarity Index Measurement

Chapter 1

Introduction

An infection of one or both lungs brought on by bacteria, viruses, or fungus is known as pneumonia. The infection is severe and causes pus and other liquid to fill the air sacs. One or more lung lobes may be affected by lobar pneumonia. The extent of diagnosis greatly determines the success of treatment and management that will be given to the patient. Chest X-ray is a widely applied method of diagnosing pneumonia, though it may be difficult for the radiologists to analyze such images and it may require much time, especially if they work in areas with the limited accessibility to the Healthcare services.

Applications of AI and deep learning in the field have been receiving much attention in the past few years because of their ability to take over the tedious and time consuming nature of medical image analysis. CNNs are one of the several types of DL models, found highly effective in image recognition. They can also learn and select relevant features from images without any human intervention and can be applied to technical problems like pneumonia detection.

But the primary concern of training and deploying deep learning models for processing medical images is computational demands. Computation on traditional CPUs is often time, consuming and less efficient particularly when processing big data and using complex models or algorithms. This can be problematic for the use of AI systems for diagnostics, especially in areas where quick analysis of data is important. To overcome these challenges, we take advantage of CUDA – an umbrella for parallel computations in NVIDIA graphic processing units for enhancing CNNs for use in identifying pneumonia. CUDA introduces the ability to work parallel on GPUs (Graphics Processing Units), which speeds up both the training process and inference of deep learning models. This is important for the real time analysis, especially in the clinical setting, which is the main area of application of ECG signal acquisition.

Here in our proposed study, a novel deep learning model was created to distinguish bacterial-viral pneumonia from normal and non-variant pneumonia cases using chest X-ray pictures. We then incorporate various state-of-the-art and specifically optimized CNN architectures for feature extraction and classification. Especially through the use of CUDA for parallel processing, the computational time is significantly reduced and therefore, the proposed framework can also be applied in

real-time settings. Moreover, for performance assessment of the developed CUDA-optimized CNN model, we performed numerous experiments. The findings reveal that our model has considerable efficacy in detecting pneumonia while maintaining significantly less computational complexity as compared to others in practice. The results of this study reveal how applying deep learning and high performance computing may improve medical diagnostics in low resource environments.

1.1 Poblem statement

Even with available and sophisticated imaging methods, the timely and accurate diagnosis of pneumonia is still a major problem for healthcare since it is more common among children in the developing world. Pneumonia is an inflammatory lung disease which results from a bacterial, viral, or other pathogen, and as such, there is a need to diagnose it in the shortest time possible as well as categorize it properly to ensure that it receives the right attention from a clinician with the best results for the patient. However, the manual reporting of these CXRs requires radiologists, and this can be extremely time-consuming or rather inaccurate in cases where radiologists are overwhelmed by numerous scans or in developing countries where there is a scarcity of radiologists.

Recent advances in the DL approaches to solve this problem have proven quite beneficial but their computational limitations remain a drawback. An expensive data processing, as well as intensive computing needed for CNNs training and application in CPU-based systems are significantly challenging. This limitation greatly reduces the online analysis feature that is very important in clinical situations where decisions have to be made in real time.

Moreover, the existing models tend to classify the provided image into either having pneumonia or not, but it is beneficial to make a fine distinction between the types of pneumonia. The absence of more differentiated classification additionally hinders the applicability of these models in clinical work where identification of what kind of pneumonia can help to choose an adequate treatment strategy. For these reasons efficient diagnostic tool is needed with high accuracy and real time results to distinguish among different pneumonia types by analyzing chest X-ray images. Using GPU acceleration with NVIDIA CUDA seems like a sensible way to solve computational-intensive issues, thereby increasing the speed factor and optimizing the usage of deep learning models. The main aim of this research is to design and

train a specific CNN model capable of speedy and efficient recognition of pneumonia and other conditions such as bacterial and viral pneumonia as well as no pneumonia using Chest X-ray images. To this end integrating CUDA for parallel processing is proposed so as to improve the computational speed and scale-ability of the given model in order to facilitate the implementation of this model in real world clinical environments. This approach aims to develop a valid diagnostic aid that can be utilized by physicians to help diagnose and care for the diseased, hence promoting positive results among the populace.

1.2 Research's Objectives

The primary aim of this research is to develop an advanced, CUDA-optimized Convolutional Neural Network (CNN) model for the accurate and efficient detection and classification of pneumonia from chest X-ray images. The specific objectives of this study are:

- **Develop a Custom CNN Architecture:**
  - This would require formulating and implementing a specialized CNN that is optimized for pneumonia detection and can differentiate between the bacterial, viral, and non-pneuma types.
  - It helps to enhance the proposed model with the state-of-art deep learning approach on both the feature extraction and the classification part.
- **Leverage CUDA for Performance Optimization:**
  - Integrate NVIDIA CUDA technology to Improve the model's numerical performance by implementing the trained model on GPUs so as to shorten the overall training and inference time.
  - Optimize the model's computational efficiency by exploiting the parallel processing capabilities of GPUs, thereby reducing training and inference times.
- **Evaluate Model Performance:**
  - Perform thorough trials to determine the said CNN model's accuracy, precision rate, recall, and computational complexity.
  - The following evaluation benchmarks need to be used to assess the performance of the CUDA-optimized model against non-CUDA optimized models and similarly existing state-of-art models:
- **Implement Real-time Analysis Capability:**
  - It is crucial to formulate a set of rules that would allow the model to identify pneumonia in real-time and distinguish between different types of pneumonia, so that it could be implemented in clinical practice on time.
  - Validate the usability and effectiveness of the model in real-time clinical scenarios in order to run it virtually with practical efficiency.
- **Analyze and Interpret Results:**
  - Examine the outcomes of the experiment that have been carried out in order to determine the potential of and flaws in the proposed model.
  - Interprethow the obtained results can shed light on the effects of CUDA optimization onto the model and potential clinical applications of the developed technique.

1.3 Thesis Contributions

- **Optimized CNN Architecture for Pneumonia Detection:** We proposed a new efficient Convolutional Neural Network (CNN) architecture that can be used specifically for pneumonia detection on medical images. The embedding model in this architecture has been considerably altered to enhance feature representation which results in improved diagnostic accuracy.
- **CUDA-Accelerated Training and Preprocessing:** We used CUDA c++ to accelerate CNN training and medical image data pre-processing steps to make training happen as quick and efficient as possible. This drastically shortens the training process and is able to provide real-time pneumonia prediction.
- **Real-Time Pneumonia Detection System:** Using the optimized CNN model and CUDA acceleration, we implemented a real-time pneumonia detection system. The system instantly analyzes the images of the chest X-rays (about half a dozen in total), leading to a definitive diagnosis of pneumonia, their diagnosis could help detect the disease quickly and early treatment.
- **Enhanced Efficiency and Accuracy:** But these methods are relatively powerful in many ways when compared to the conventional methods, in particular in terms of better computational efficiency and the more reliable diagnostic accuracy[5]) This well optimized model offers an efficient tool for the timely and accurate detection of pneumonia with deep learning, together with GPU acceleration to provide rapid and reliable results. This development has tremendous promise for improving medical image analysis and assisting healthcare providers in clinical purposes.

So, this research work boost the field of medical image analysis by presenting an unique and efficient real time pneumonia detection system that uses CUDA acceleration and for its CNN architecture with a unique embedding structure. Although this is one of the most common diseases, it has the potential to help with faster and more correctly diagnosing pneumonia, thus improving patient care.

1.4 Thesis Structure

Chapter 1: This is the introduction chapter where the researcher introduces the topic of study, gives a background to the study, outlines the research objectives and questions, and justification for the study. Chapter 2: This is the literature review chapter where the researcher provides an overview of the different literatures that have dealt with the topic of study to avoid repetition. Chapter 3 – This is the theoretical framework chapter where the researcher introduces Beneath the chapter title "Introduction", the first chapter addresses: problem statement, research objectives and thesis contributions. Chapter two of the book is titled as "Literature Review" which gives information on pneumonia, Convolutional Neural Networks (CNNs), Compute Unified Device Architecture (CUDA) and works of different researchers.
In the chapter four, titled "Dataset Description", the current paper presents the information regarding the chosen dataset, data pre-processing as well as data augmentation utilized in the research. In the fourth chapter, titled "Methodology,

Architecture, and Model Specification," It presents the assessment of conventional existing models, the appreciation of Custom Siamese Networks, and the formation of a highly effective tender model, including the Siamese embedding model and the dedicated CUDA sliding window L1 distance layer.
Chapter five, titled "Result Analysis," provides a discussion on the evaluation of the proposed model, utilised evaluation parameters, comparison of distance layers, as well as confusion matrix. Last but not the least, Chapter six carries the title, "Conclusion," which restates the findings of the study, the conclusions drawn in relation to the research question and hypotheses, the realization of the challenges faced during the research and the approaches attempted to minimize these issues.

Chapter 2

Literature Review

2.1 Background

2.1.1 Pneumonia

Pneumonia is a widespread respiratory illness that manifests as an acute infection and causes immense morbidity worldwide, with children under five years and older persons being most affected. Pneumonia, WHO estimated that about 15 % of childhood deaths were within under five years worldwide, and this causes about 800000 children deaths annually [6]. It can be caused by bacterial and viral infections, fungal infections, the flu, pneumonia, and COVID-19, but some symptoms include coughing, fever, and difficulty in breathing.
Infermity and mortality: Pneumonia is not only a children disease it is also fatal among the elderly and immunocompromised who frequently suffer severe morbidity and are at high risk of dying from the disease globally. This stresses the significance of timely and correct diagnosis since further management could worsen if the condition is left unaddressed can result in dire consequences and high mortality rates. The integration of enhanced diagnostic methods including the deep learning models and imaging has enabled the ideal identification of required diagnostic outcomes and determination of patients' status.
Especially in developing countries where access to healthcare is relatively low, the burden of disease from pneumonia is fearfully high noting the need to develop effective and accessible diagnosis techniques. Pneumonia poses a large global health threat; campaigns coaching the value of vaccination, prompt care-seeking, and creating a new generation of diagnostic tools would be relevant.

2.1.2 Convolutional Neural Networks (CNNs)

CNNs are deep learning models designed to work with structured in grid data particular images for computation. They have in fact claimed that they have attained the state-of-the-art performance in the general range of computer vision applications including image categorization, localization, and partitioning. CNNs resembled the architecture of the animals' visual cortex and are designed with several layers for layer-wise learning of the spatial hierarchy of features from the input images in an automatic as well as adaptive manner. The CNN architecture contains three main levels; they are the convolutional, pooling, and fully connected level. A convolution

layer uses a kernel or filter to convolve the input image and produces features maps of its input that help in defining local patterns like edge, texture, shapes etc. To explain this, the filters are placed on top of the input image, and vector multiplication by corresponding elements occurs – helpful for training spatial features. Batch normalization scales and shifts the activations with learnable parameters so that the preceding layer's outputs are normalized; activation functions such as ReLU (Rectified Linear Unit) which introduces nonlinearity into the data to enable learning more complex patterns.
Stride layers, commonly, max-pooling layers, downsample the feature map and thereby reduce the computational dimension and the parameters. This process also helps in making the model translation invariant with small transformations with the input image. The last one is in the last layer where industry application is completed and the first two are composed of convolution and pooling layers feature map.
CNNs use various methods in improving it performance these includes; Batch Normalization, Dropout and Data Augmentation among others. Batch normalizes and hastens up the training process and also fight off the issue of internal covariate shift by normalizing the input to that layer. L2 regularization helps to address the issue of overfitting which is a scenario whereby, some of the neurons are intentionally 'dropped' while training in a bid to enable the network learn better features. This mainly entails leading the existing training images through some processing mechanism like the rotating, scaling and flipping which in one way or the other assist the model to enhance its capacities of generalization.

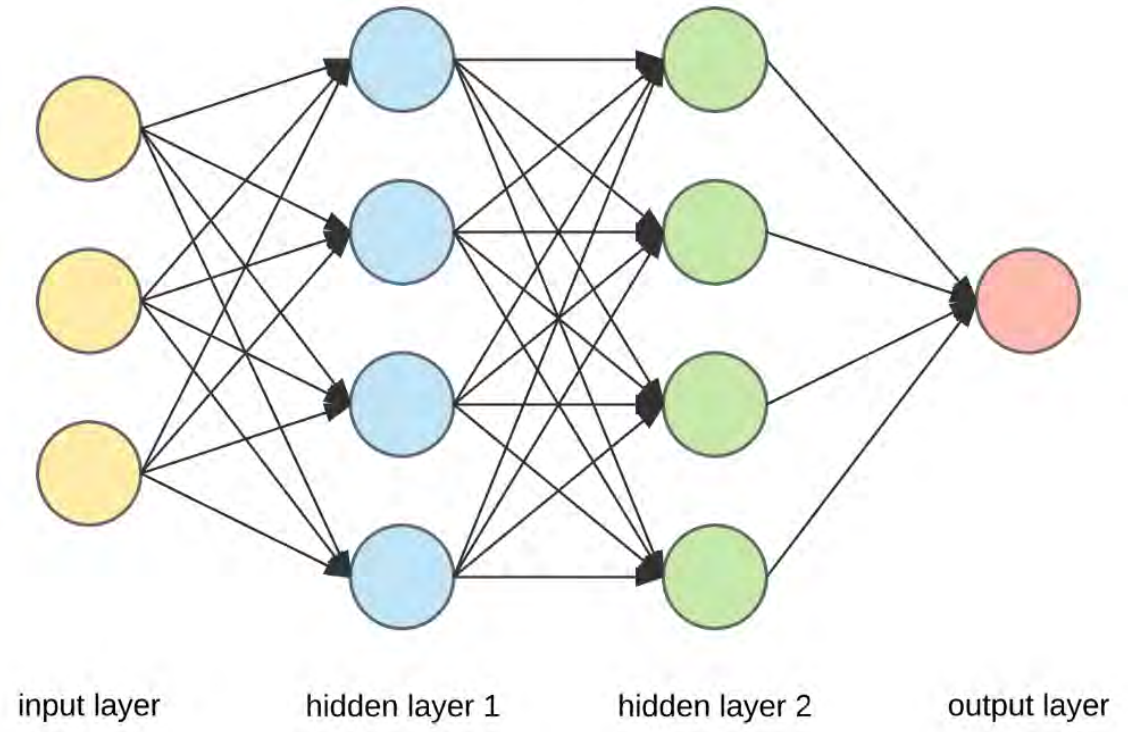

Figure 2.1: Convolutional Neural Networks

2.1.3 Compute Unified Device Architecture (CUDA)

Compute Unified Device Architecture or CUDA [3] is a parallel computing platform and application programming interface (API) which is created and developed by Nvidia. It is a niche and conservative field of programming which allows developers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing that is known as GPGPU. It is an incredible environment for High-performance computing (HPC) yet due to its capacity to scale, it is perfect for Deep learning, Scientific computations, and Real-Time Processing functions.

- **Global Memory (DRAM):** Huge, NFS access by all threads, but slow.
- **Shared Memory (L1 Cache):**Faster and shared between threads in the same block.
- **Constant and Texture Memory:** Specialized memory spaces for read-only data
- **L2 Cache:** Acts as a high-speed data storage, used by the CPU.
- **L3 Cache:** Enhances Data Access Speed across Mutliprocessors

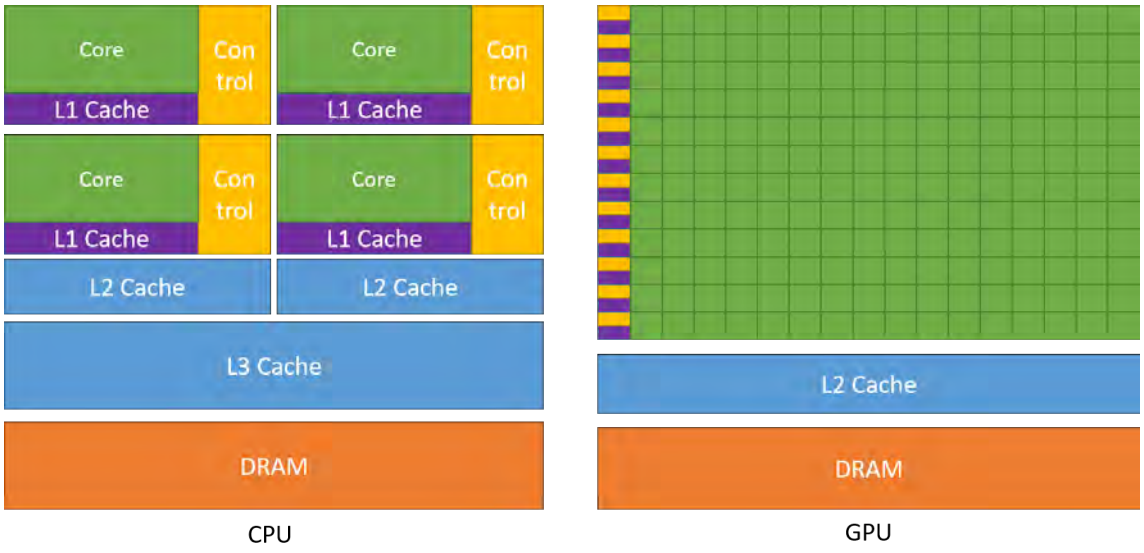

Figure 2.2: CUDA Architecture: Hierarchical Memory Structure

The figure above illustrates the hierarchical memory structure which showcase the separation of caches and control units that enhances computational efficiency and memory management.

Advantages

- **Parallelism:** CUDA allows thousands of threads to execute concurrently, speeding up the processing of computationally intensive workloads.
- **Scalability:** The architecture scales with the number of available cores which makes it adaptable to various GPU models and sizes.

- **Efficiency:** Provides data access with minimal latency and enhances overall performance by means of hierarchical memory structure.
- **Flexibility:** CUDA supports a wide range of applications; from scientific simulations to deep learning, through its comprehensive programming model.
- **Community and Ecosystem:** A strong pool of libraries, resources and support in community aids in development and optimization.

Applications

CUDA has changed the game in many fields, particularly in the area of deep learning where it speeds up the training and inference of neural networks. It is used in medical imaging to allow for real-time processing and analysis, essential for tasks like the detection of pneumonia from X-ray images.

2.2 Related Works

The paper titled, 'Deep Learning for Automatic Pneumonia Detection' by [15] provides a detailed and fairly exhaustive understanding of how the concept of deep learning can be applied to diagnosing pneumonia from chest X-ray images. One of the major objectives in the existing research is to try out different types of CNN structures and approaches toward achieving higher diagnostic performance. It is also important to note that the authors use the feature extractor models like transfer ability of VGG16, ResNet50, and InceptionV3 trained on the dataset of the pneumonia image. Based on the findings, deep learning models, particularly when applied with transfer learning, are much more effective than conventional approaches in diagnosing pneumonia, indicating the ability of the methods in supporting radiologists in or outpatient practice. This study demonstrates that the improvement of the structures of neural network and the introduction of data augmentation techniques are critical factors in enhancing the accuracy and reliability of Medical Imaging applications.

The paper entitled "An Efficient Deep Learning Approach to Pneumonia Classification in Healthcare" by Okeke Stephen published in 2019 [14] proposes a new pneumonia CNN model from scratch that was trained to detect pneumonia based on chest X-ray images. The problem with deep learning is that at times it can be highly demanding in the amount and quality of data to generate and so unlike the traditional methods that just use transfer learning, this study uses data augmentation techniques to try and increase the accuracy of the model with the little amount of data available. The employed CNN architecture which is fine-tuned to the task of feature extraction and classification implies high validation accuracy. It is equally important also because the accuracy of using this approach in diagnosing pneumonia is commendable, especially for health facilities that are poorly endowed. The

paper "CheXNet: The paper titled "Detecting pneumonia from chest X-rays using deep learning" by[10] develops a more enhanced 121 layers Convolutional Neural

Network (CNN) named CheXNet to detect pneumonia in the chest X-ray scans. By using DenseNet structure, the proposed model was trained on ChestX-ray14 dataset which includes over 100, 000 X-ray images labeled with 14 classes of thoracic diseases. CheXNet also adopts the transfer learning to improve its performance and retrain the network just for the detection of pneumonia. Using the F1 score as the measure of accuracy, the given model was able to establish an f1 of 0. 435, which was even more accurate than practicing radiologists assigned to aid in the study. This work reveals that deep learning is well developed for the medical diagnosis and can bring significant changes to enhance the diagnosis speed and accuracy. Large-scale annotated datasets, deep learning architectures and pre-trained models are also emphasized as critical elements for building reliable, generalizable AI tools that can be adopted in clinical practice settings.

The article by Ali Bakhoda [3] provides extensive detail about the SIMD or SIMT model, which Cuda uses. The SIMT model used by Cuda stands for Single Instruction Multiple Thread, where multiple threads work the same based on one single instruction. But they proceed on various data points, and all are being calculated parallelly. Furthermore, this work describes a large number of non-graphic programs developed in the Cuda model running on a new microarchitecture powered by the parallel thread execution (PTX) virtual instruction set from Nvidia. The author also claims that the number of GPU cores is much more than the CPUs, which is really helpful for the parallel tasks used in various machine learning or deep learning models.

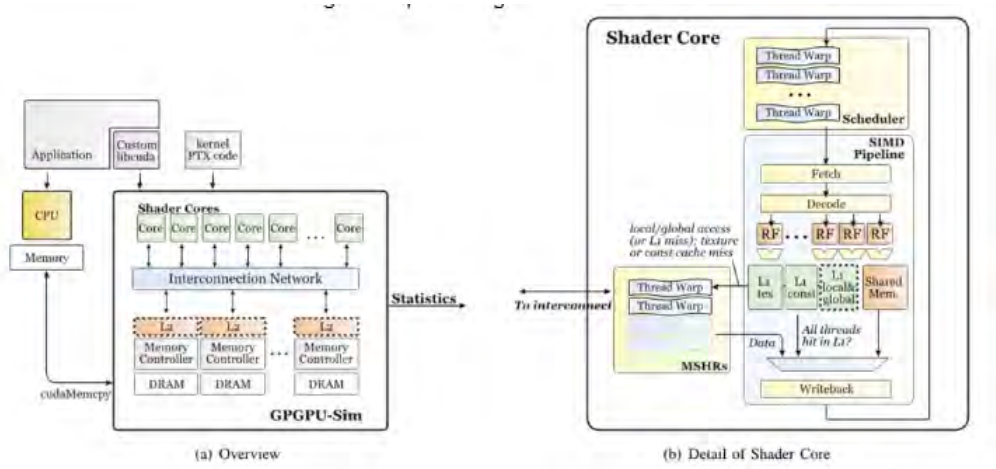

Figure 2.3: Cuda Architecture [3]

The research article "Diagnosis of Pneumonia Using Deep Learning" by [12] examines the ability of deep learning to detect pneumonia and classify chest X-ray images to normal, bacterial, or viral pneumonia. The authors used CNN to build a model that: can accurately differentiate between bacterial pneumonia, viral pneumonia, and normal cases. The study establishes CNN as highly accurate and efficient in medical diagnosis, which forms the foundation of deep learning. To increase the efficiency of the algorithm, data augmentation and preprocessing strategies were used. From the results presented in the model, the accuracy was found to be very

high, thereby implying the ability of the model to support quick and more accurate diagnosis of pneumonia by healthcare practitioners.

The development of mainstream software that can take use of the growing number of processing cores in multicore CPUs and many-core GPUs is covered in the paper by R Farber [5]. This is basically done by Nvidia's Cuda architecture and implements the microarchitecture in scalable parallel systems. Additionally, Stratton et al.'s framework, known as "prototype source-to-source translation," which maps a thread block to loops within a single CPU thread in order to compile CUDA programs for multicore CPUs, is explained in the study. Recent GPGPU models are comparable to the kernels used by Cuda. It differs, though, in that it offers thread blocks, shared memory, global memory, and flexible thread generation.

The SIMD model of Cuda is described in the paper "Accelerating Large Graph Algorithms on the GPU Using CUDA" by Pawan [1], which also explains how several GPU threads can work on a single instruction. SIMD, or single instruction multiple data model, is vastly used in machine learning and ai sectors to work with different parallel data points. In this paper, an implementation of a large graph was shown involving almost a million vertices. Additionally, the author demonstrated how cuda performs incredibly well on a few standard algorithms, including all-pairs shortest path, single source shortest path, and breadth-first search. Here from the table we can see how efficiently cuda cores executed the tasks and it takes much less time when the graphs are not linear. The cuda cores are unable to achieve optimal performance when the graph is linear because each loop requires processing every vertex, which lowers speed.

	Number of Vertices	Number of Edges	BFS CPU time(ms)	BFS GPU time(ms)	SSSP CPU time(ms)	SSSP GPU time(ms)
New York	250K	730K	313.117	126.04	1649.85	760.14
Florida	1M	2.7M	1055.22	1143.99	7357.83	7906.49
USA-East	3M	8M	3844.35	4005.75	27000.2	35777.52
USA-West	6 M	15M	6688.78	7853.19	48814.4	63749.54

Figure 2.4: Cpu vs Gpu computation comparison [1]

Michael Garland's research article [2] describes the architecture and operation of CUDA as well as the simple implementations of the SAXPY procedure that are defined by the BLAS linear algebra library. serial implementations on a CPU calculate one element in each iteration while all these independent elements are being computed in parallel, assigning each a separate thread. The paper describes the Tesla unified graphics architecture designed by Nvidia to accelerate parallel programming. From the figure, we can see how well the GeForce 8800 + Core2 Duo computes comparing doing a task only with the help of Core2 Duo (CPU). This is because all the parallel tasks are being done in the Cuda kernels, and the complex serial tasks are done in Cpu, and both of the hardware is being used at the same time.

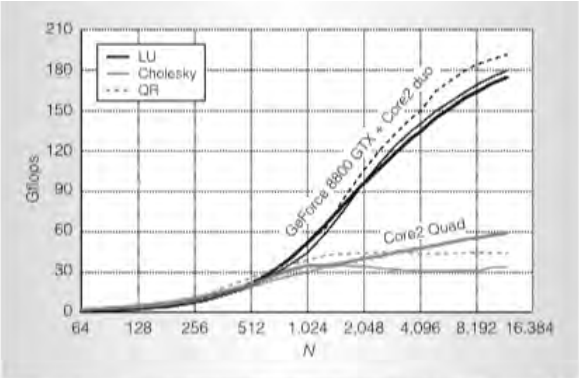

Figure 7. Performance on dense LU, Cholesky, and QR factorization for **N** × **N** matrices.

Figure 2.5: performance on dense LU [2]

A GPU parallelization technique for 3D finite difference stencil computing using CUDA is described in the paper by Paulius Micikevicius[[4]], where it delivers an order of magnitude speedup over comparable seismic industry-standard algorithms. One drawback of this method is that it ignores other framework parallelization techniques in favor of concentrating solely on the 3D finite difference using Cuda. However, it also explains how to leverage several GPUs in a single system to accomplish linear scaling with GPUs through the use of asynchronous computing and communication.

The paper "Comparison and Validation of Deep Learning Models for the Diagnosis of Pneumonia" by [16] evaluates several deep learning models for classifying pneumonia from chest X-ray images. The study uses the Kaggle dataset, containing 5216 training and 624 testing images, to compare the performance of five mainstream CNN algorithms: a regular CNN, ResNet-50, MobileNet, VGG19 andResNet-18, . MobileNet, enhanced with depthwise separable convolutions, emerged as the most efficient, achieving the highest accuracy (92.79%) and recall (98.90%) with significantly lower computational costs. This research underscores the potential of lightweight CNN models like MobileNet for rapid and accurate pneumonia diagnosis, particularly in resource-constrained clinical settings.

In the paper titled "Early Diagnosis of Pneumonia with Deep Learning " written by Deniz Yagmur Urey, Can Jozef Saul and Can Doruk Taktakoglu in 2019 [13], deep learning approach was proposed for early diagnosis of pneumonia through chest X-ray image analysis. To predict the tumour size, the authors proposed the deep learning structure of the CNN and residual networks, with preprocessing steps to improve feature of images. Their approach entailed increasing the contrast of images; changing the color space of features; and adding artificial light onto features, so as to make out diagnoses more distinctly. These types of classifications attempted at attaining an accuracy of 78% as per the proposed model outlined in fig. 73 percent of the samples, a much higher rate than before, which indicates that future efforts in diagnosing early pneumonia may benefit greatly. In the current study, the importance of machine-based techniques is expressed to maintain the level of accuracy and reproducibility in the diagnoses revealing a possible shift in time that eliminates human-centered imaging interpretation in clinical applications.

Chapter 3

Dataset Desccription

3.1 Description of the Dataset

Dataset Source

The dataset for this study includes chest X-ray images from two sources on Kaggle: "Chest X-Ray Images (Pneumonia)" and "Chest X-ray (COVID-19 & Pneumonia)".

Organization

The datasets are organized into train, test, and validation folders where each contains subfolders for different categories.

Content

- **Total Images:** 5,863 chest X-ray images (JPEG format)
- **Categories:** Pneumonia, Normal, and COVID-19
- **Source:** Pediatric patients from Guangzhou Women and Children's Medical Center

Quality Control

All radiographs were screened for quality, removing low-quality scans. Diagnoses were verified by two expert physicians and then reviewed by a third expert.

Clinical Context

- **Normal:** Clear lungs without abnormal opacification
- **Bacterial Pneumonia:** Focal lobar consolidation
- **Viral Pneumonia:** Diffuse interstitial pattern

COVID-19 Pneumonia Data

- **Source:** "Chest X-ray (COVID-19 & Pneumonia)" on Kaggle
- **Additional Categories:** COVID-19
- **Total Images:** Included in the overall count
- **Content:** Chest X-rays of COVID-19 patients showing characteristic lung opacities

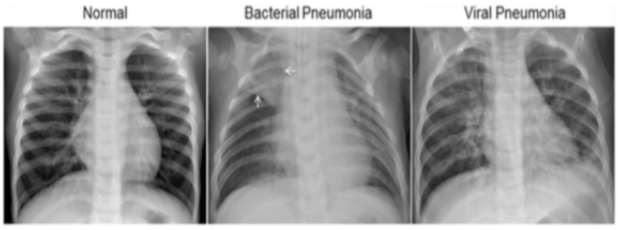

Figure 3.1: Illustrative Examples of Chest X-Rays in Patients with Pneumonia.

3.2 Data pre-processing

Data pre-processing of chest X-ray images is the foundation of the deep learning model preparation which includes several steps. First, each image is loaded using the file path and its format is changed to gray, as all input data should be unified. These images are then reduced to 100x100 pixels to ensure that the real images fed to the model are in this required size. Next, the pixel values are scaled to get a pixel value ranging from 0 to 1 by dividing the pixel value by 255 which helps in stabilizing and speed up the training of the network. This pre-processing pipeline is crucial to regulate a variation in the input data so that the probability of learning data in an optimal way and predicting data accurately is improved. If done systematically, the transformation of raw images into the right format contributes to efficiency and boosting the generalization capacity of the model in unseen data sets.

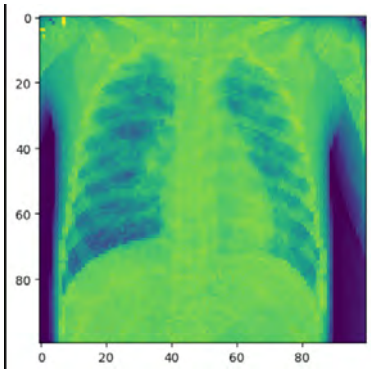

Figure 3.2: Sapmle data image after Image Pre-Processing

3.3 Class Distribution

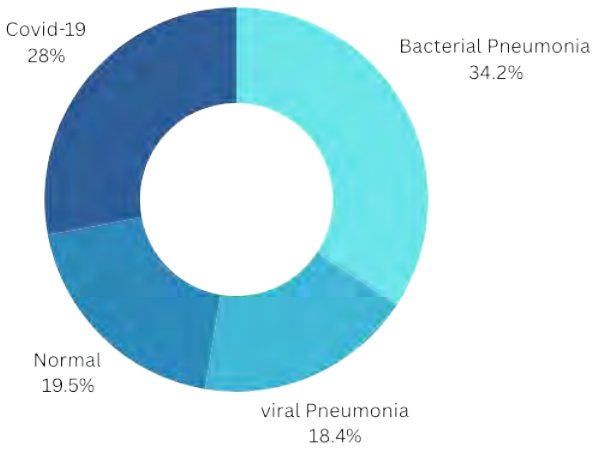

Figure 3.3: Distribution of 4 classes

In the pie chart labeled above, it is possible to get some insights into the distribution of elements of our dataset by classes which is also imbalance. Although, SNN can efficiently deal with such imbalance sample; therefore, augmentation of this imbalance sample was not performed. SNNs learn the matrices of similarities between pairs of inputs; therefore, they require no further examples for the classification of some classes of inputs. Such approach to learning from comparing with other classes, as well as from using such methods as selection of pairs for comparison and the contrastive loss gives the input SNN to train a very effective model of the classes while at the same time making sure that the integrity of the datasets is upheld.

Chapter 4

Methodology, Architecture, and Model Specification

4.1 Evaluation of Well-known Existing Models

While exploring for the best model to incorporate into our model of detecting Pneumonia we considered the following four deep learning models namely ResNet, Inception and VGG16.

ResNet (Residual Networks)

ResNet was proposed in [8] to overcome the vanishing gradient problem and employs the residual connections that are actually the connections that allow to skip one or more layers. These connection enable the network to have capability of learning for identity mappings and as a result have the ability of training very deep networks with up to hundreds or thousands of layers. As for the performance of ResNet on several benchmark datasets, ResNet yielded excellent results. For example, ResNet-50, with depth of 50 layers, reached 3% top-5 error rate. They obtain 6% on the ImageNet dataset, which is higher than ever seen before with previous models.

Inception

Inception, proposed by Szegedy et al.[9] , Completion organizations use inception modules that extract multi-scale characteristics at the same depth. The convolution operation involves applying multiple filters of different sizes (1x1, 3x3, 5x5) in the same module which gives the network the ability to learn the fine as well as the coarse features efficiently. Comparing to other deep learners, Inception is highly accurate and fast, which makes it important in large scale image classifications. For example, the Inception-v3 model described above recorded a top-5 error rate of 3%. Every time is better than other networks with accuracy of 46% on the ImageNet dataset, which again proved that it performs better and faster.

VGG16

Another popular CNN is VGG16 designed by Simonyan and Zisserman [11] mainly known for its simplicity and depth. It comprises of multiple convolutional layers

where several stacks of 3x3 filters are placed to ensure the extraction of hierarchical features; fully connected layers for classification purposes. The presented approach of the model's architecture and the increased depth of the VGG16 network allowed achieving high results in image classification. Surprisingly, VGG16 proposed a top-5 error rate of 7.3 percent despite its rather basic structure. 3% on the ImageNet dataset, but it has more than 138,089,840 parameters that are making it very large, complex and computationally intensive.

4.2 Preference for Siamese Networks

Despite the robust performance of existing models like ResNet, Inception, and VGG16, we found the concept of Siamese networks particularly compelling for our application. Here, we explain the advantages and reasons for preferring Siamese networks over these traditional models.

- **Similarity Learning:** Siamese networks are explicitly designed to learn a similarity function, which is crucial for distinguishing between similar classes. In medical image analysis, such as distinguishing types of pneumonia from X-ray images, subtle differences need to be identified accurately. While ResNet and DenseNet are excellent at feature extraction due to their depth and residual connections, they primarily focus on classification tasks rather than measuring similarity. In contrast, Siamese networks excel in verification tasks, making them ideal for medical diagnostics where understanding the relationship between pairs of images is critical.

- **Pairwise Comparisons:** Siamese networks operate by comparing pairs of images, learning to determine whether the images are similar or different. This approach enables the network to generalize better to new, unseen data. Inception and VGG16, though powerful, are designed to extract multi-scale features and hierarchical features respectively, and are not optimized for pairwise comparison tasks. The pairwise comparison capability of Siamese networks allows them to leverage smaller datasets more effectively by focusing on the relationship between image pairs rather than solely on individual image classification. This is particularly advantageous in medical fields where labeled data can be scarce and obtaining more labeled data is costly and time-consuming.

- **Robustness to Class Imbalance:** Siamese networks handle class imbalance effectively by focusing on the relationship between pairs rather than the individual class distributions. Traditional models like ResNet, DenseNet, Inception, and VGG16 can suffer from performance degradation when trained on imbalanced datasets. In medical image datasets, certain conditions (e.g., rare diseases) may have significantly fewer examples compared to common conditions. By learning to discriminate based on similarity rather than absolute classification, Siamese networks mitigate the impact of class imbalance, leading to more reliable and balanced performance across different classes.

- **Efficiency and Practicality:** Where previously proposed ways need lots of data with human annotation of each class, the Siamese networks are getting

trained on less no of samples by focusing on pairwise comparisons. By administering tests using motion data to verify each implementation, it can reduce the amount of data to be collected and facilitate faster development. Moreover, Siamese networks are designed in such a way that it allows us to integrate new classes for recognition without much headache of forward pass complexity, which is extremely beneficial in fields like medical diagnostics where addition of new signs, image for diagnosis is a routine affair.

Limitations of Siamese Networks:

While Siamese networks offer significant advantages; they also come with certain limitations which must be considered:

- **Computational Intensity:** Siamese networks require lot of computational resources to train. This is because there are pairs of images input in the model instead of single images, this doubles the amount of data to be processed. During training, for every pair of images, the embeddings are calculated for both and a similarity measure between them is computed. This adds a computational burden, which means that you can wait much longer for the training times and sometimes this is not an option, and for that you need to have hardware reinforcement like GPUs. This requirement of substantial computational power is a hindrance, especially in areas deprived of enough resources.
- **Classical Siamese Networks and Binary Classification:** Traditional Siamese networks are mainly used for simple binary classification problems where the model has to decide if two inputs belong to the same class or not. This limitation might be a disadvantage in cases where tasks that involve multi-class classification, that is, tasks that involve making a forecast of which of the multiple possibilities is the correct class. Applying the Siamese networks towards multi-class classification brings extra difficulties and research into the design since the primary Siamese network is a binary classifier, and other techniques like triplet loss need to be applied to enlarge upon the network's capability.

4.3 Development of an Efficient Custom Model

To overcome these limitations, we redesigned the model that is based on the Siamese network, though with certain modifications. To adopt and keep the learning similarities' idea present in the original model; the custom model proposes a newly developed embedding layer and a main structure model. The new model is designed for improved feature extraction and classification accuracy.

4.3.1 Embedding Model

We analyzed and examined the architecture of the Siamese network from the paper "Siamese Neural Networks for One-shot Image Recognition" by [7] source. It is also important to note that the paper has a clear breakdown of how the architecture of the Siamese network was analyzed and examined. It was this paper that came up with the concept of implementing Siamese networks in the form of image recognition

with special focus on their capacity for learning similarity measures. In this case,

the embedding model includes several convolution layers with max pools after each convolution layer. It should be noted that in the present case convolutional layers are used for detecting a range of features at the varying level of abstraction. The first layer would identify edges and textures that are basic to image recognition whilst the subsequent layers could identify intricate features and structures in the image. This is generally achieved through a hierarchy of feature extraction functions that help the model capture high dimensional and informative representations of the input images.

- Initial Layers: Focus on low-level features like edges and textures.
- Intermediate Layers:Capture mid-level features such as shapes and patterns.
- Deeper Layers: Detect high-level features like object parts or larger structures.

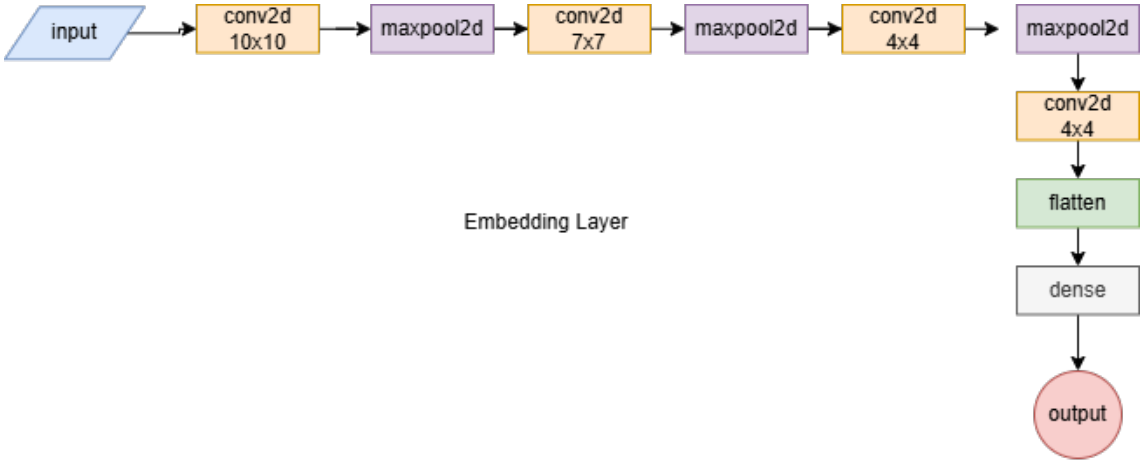

Figure 4.1: Embedding Architecture by [7]

Complexity

As mentioned earlier, architectures with this form of capability are very useful in obtaining comprehensive features. But it tend to be computationally intensive and therefore very slow. The convolutional layers are known to consist of many parameters that are unique and are required to be learned and this enhance the computational cost. Max-pooling layers are used to decrease the number of dimensions which also leading to increased computational time. Training such a model calls for massive I/O requirements for data handling through high end graphics processing units, GPUS as well as take a very long time which makes the model less suitable for real time applications specially in those cases where high end resources are very unlikely to be available.

Limitations

The high computational cost of the embedding model poses several challenges:

- **Resource-Intensive:** Relies on a fair amount of processing and memory units which may need internal and external support in most deployment scenarios.
- **Slow Inference:** Through the model complexity there comes a disadvantage of slow computation which is unhelpful to any real life problem where decisions are needed instantly.
- **Scalability Issues:** Sometimes, such increases can be extremely challenging especially when dealing with large data sets or when the model has to be scaled to handle more classes, in such scenarios it becomes cumbersome for some actual applications.

Proposed Embedding Architecture

Our updated mapping has five convolutional blocks and each of them is designed to discover progressively more abstract representations of the input images. Here's a detailed explanation:
Architecture:

- **First Block:** The initial convolutional layer has 32 filters. Here each of them forms with a 5x5 kernel, followed by batch normalization and max-pooling.Mainly, This layer captures basic features like edges.
- **Second Block:** The initial convolutional layer has 64 filters with a 3x3 kernel, followed by batch normalization and max-pooling.Mainly, captures more complex patterns.
- **Third Block:** The 3rd convolutional layer has 32 filters with a 3x3 kernel, followed by batch normalization and max-pooling.Mainly,capture intermediate-level features.
- **Fourth Block:** Here, total 256 filters with a 3x3 kernel are applied, followed by batch normalization and max-pooling. And this basically captures even more complex features.
- **Fifth Block:** The final convolutional layer has 512 filters also with a 3x3 kernel, followed by batch normalization and max-pooling. And this mainly to capture the most abstract features.
- **Final Embedding Block:**Finally, The feature maps are flattened and passed through a dense layer with 1024 units with a sigmoid activation in order to generate the final embeddings.

Advantages:

- **Efficiency:**Using batch normalization, the learning process of the model becomes more stable and it helps the model to converge faster and hence reduces the time taken for training the model.
- **Regularization:** Max-pooling layers help in reducing overfitting by down-sampling the feature maps and then regularizing the selected features.

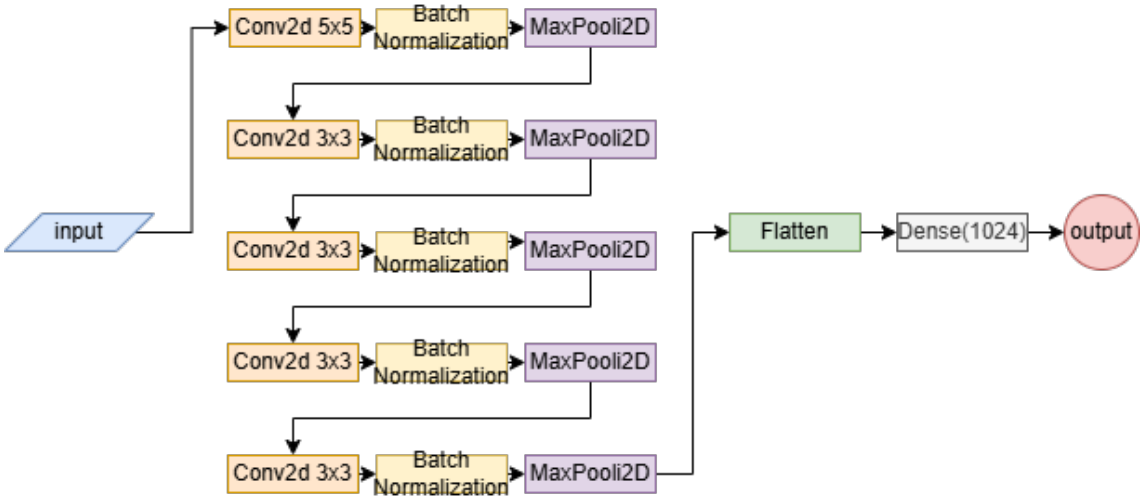

Figure 4.2: Proposed Efficient Embedding Model

- **Layer Depth:** The model is able to learn features from multiple stages of abstraction and identity hierarchical features, making it better at finding more nuanced patterns than what was possible with previous architectures.
- **Computation:** Even though the depth is more, using smaller kernels (3x3), and efficient pooling layers together do not make this Siamese version very computationally expensive unlike conventional Siamese networks.

4.3.2 Custom CUDA L1 Distance Layer

We introduce a custom CUDA L1 distance layer that combines the raw power of CUDA C++ with Python in order to perform accelerated L1 distance calculations on GPU. This consists of writing a CUDA kernel to handle element-wise computations and exposing it over a Python interface for easy integration with PyTorch.
Implementation:

- **CUDA Kernel:** The CUDA kernel (`l1_distance_kernel`) computes L1 distance by subtracting corresponding elements of two tensors and taking the absolut value.
- **PyTorch Extension:**handles out memory for the output tensor, launching the CUDA kernel, and synchronization.
- **Python Interface:** This Function is exposed in Pure C++, using PyTorch C++ extension API as a Python Module For direct Call into PyTorch Models.

Benefits:

- **Efficiency:**
  - **Speed:** :As opposed to CPU-based implementation the CUDA kernel applies thousands of GPU cores to perform parallel computations on large tensors and takes less time for the computation.

  - **Resource Utilization:** Moreover, Offloads computations to the GPU and relieves the burden from the CPU and benefits the system's capability of managing multiple applications.

- **Scalability:**

  - **Large Datasets:** The CUDA implementation works well with large size of datasets, and therefore is suitable particularly for high dimensional data format that is commonly used in the deep learning.
  - **Real-Time Applications:** The rate of the GPU computations meets the real-time processing requirement . And it is crucial within applications like the medical image analysis which needs immediate results.

Therefore, by utilizing this custom CUDA L1 distance layer, the model achieves significant improvements in computation speed, scalability, and overall performance. As a result, makes it an ideal choice for complex deep learning tasks requiring efficient distance calculations.

4.3.3 Final Proposed Model

our proposed model runs on a dual-stream architecture that is designed to distinguish between X-rays that match and those that don't. Two different kinds of image pairs are processed by this complex model where input images paired with matching positive images and input images paired with non-matching negative images. Moreover, the model structure is capable of multi-label classification. This architecture extends traditional Siamese networks by incorporating a custom embedding layer, a distance layer, and a softmax classification layer which enables it to classify inputs into multiple categories efficiently.

- **Two Streams of Flow:**
  - **Input and Validation Image Processing:** In the first stream, the input image is transformed and in the second stream, the validation image is transformed. Both images go through the same CNN network layers in getting the feature embeddings of the pictures.
  - **Embedding Comparison:** The embeddings obtained from both the streams are then compared using the intra-router L1 distance layer which has been specially designed and implemented in CUDA to calculate the Manhattan distance between the two embeddings.
  - **Multi-Label Classification:** The distances calculated during the distance layer are taken into a dense layer with softmax activation for the choice of input by the model in several categories, for example, various types of pneumonia or normal conditions.
- **Embedding Layers:**
  - **Input Processing:** In this model, two sets of identical convolutional neural network units are applied to analyze the input and validation images, capturing high levels of deep features.
  - **Convolutional Blocks:** Every chosen CNN includes 5 convolutional layers, each of which is succeeded by the batch normalization layers and the max-pooling layers. This hierarchical feature extraction abstracts out or captures all the details that are important for proper classification into the different classes.
  - **Flattening and Dense Layer:** The embeddings of the convolutional blocks are flattened and the dense layer is applied with 1024 nodes and sigmoid activation function to produce the final representations.
- **Distance Layer:**
  - **L1 Distance (Eucladian Distance):** A fully differentiable CUDA-accelerated L1 distance layer computes the difference between embedding vectors of input and validation images and takes the absolute value. Despite the presence of other distance metrics, Euclidean distance is used for its simplicity and suitability for measuring similarity, which is crucial for comparing medical images and identifying small differences between the two images.
- **Classification Layer:**
  - **Softmax Activation:** The last layer of the classification layer is a dense layer with softmax activation which can generate probability of each class that the model is trained on. This configuration enables Multiple-label classification which distinguishes different types of pneumonia and normal person's chest X-Ray.

Loss Function:

- The model uses categorical cross-entropy as the loss function to handle multi-class classification tasks effectively.

Optimizer:

- The Adam optimizer is used for training the model which provides adaptive learning rate optimization in order to enhance convergence speed and accuracy.

Training Process:

- **Data Preparation:** The dataset is divided into training, validation, and test sets to ensure model evaluation.
- **Model Training:** The model is trained using the training set, with the help of loss function and optimizer that guides the learning process.
- **Validation and Tuning:** The validation set is used to tune hyperparameters and avoid overfitting which ensures the model's generalization well to unseen data.
- **Evaluation:** Finally, the model's performance is evaluated on the test set by assessing metrics such as accuracy, precision, recall, and F1-score in order to ensure its effectiveness in real-world applications.

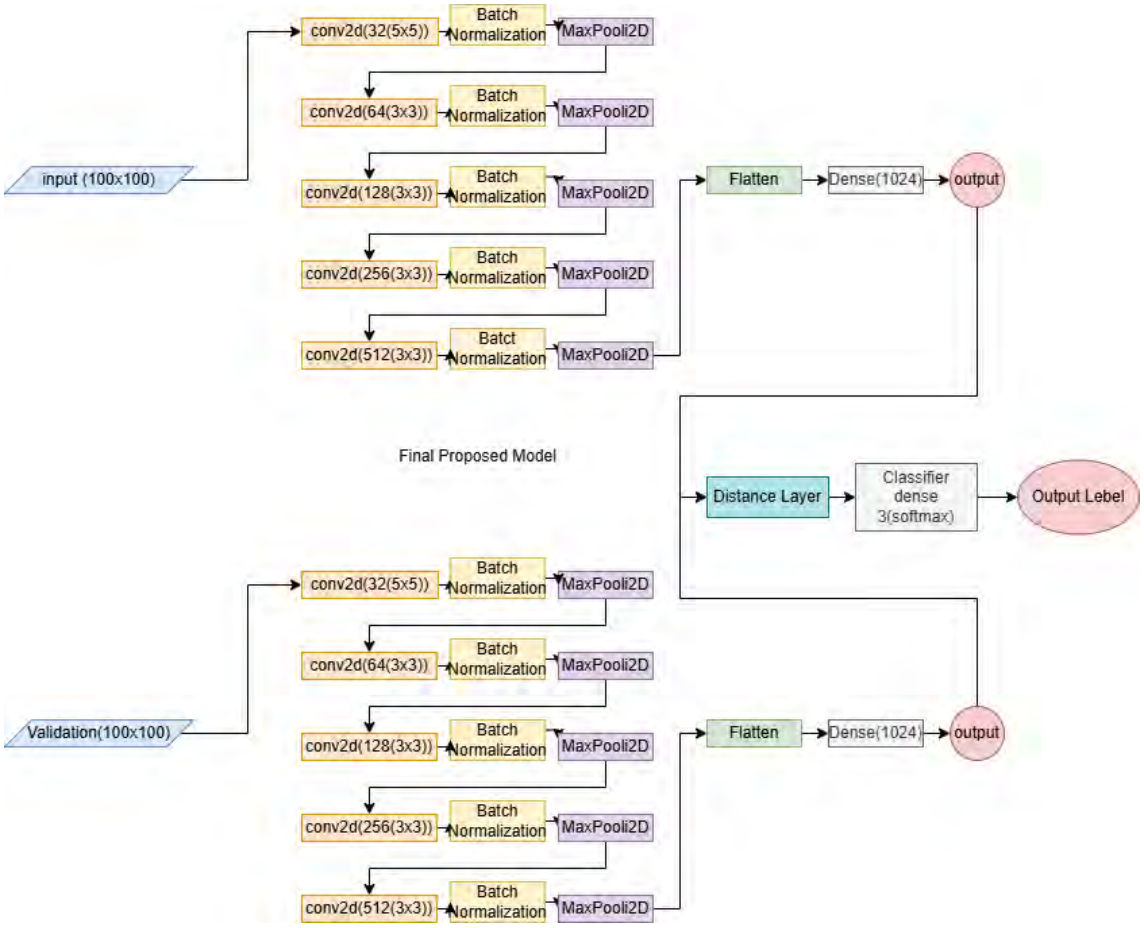

Figure 4.3: Final Proposed Model

Chapter 5

Result Analysis

5.1 Performance Assessment of the Proposed Model

We perform the assessment test in order to test the performance from our proposed model and the outcome shows excellent performance in all types of tests. The proposed model accurately differentiates between various forms of pneumonia and demonstrates satisfactory results in terms of the model's accuracy, recall, and precision.

Evaluation Metrics

- **Accuracy** `By observing the accuracy rate achieved it was evident that a high number of test samples were grouped properly by the model.`
- **Recall** `This is vital in medical diagnosis and revealed the model has a high TRUE POSITIVE rate for pneumonia and hence, few FALSE NEGATIVE.`
- **Precision** `The precision metrics depicted the capability of the model in filtering the actual positive results and eliminating chances of accidentally predicting false positives.`

Efficiency The model is designed in such a way that it effectively analyses images and uses optimized layers and CUDA cores to provide real-time performance. This efficiency is very beneficial in real-world applications that are pertinent in medical facilities where rapid identification of diseases is crucial.

5.1.1 Evaluation Metrics of proposed model

The proposed model demonstrated superior performance metrics on both the training and testing datasets, with high recall and precision values indicating its effectiveness in pneumonia detection. Table 5.1 shows the detailed performance metrics.

Metric	Value
Recall	0.9902915
Precision	0.9906344
Mean Squared Error (MSE)	0.0020542317
Pixel Accuracy	0.99063

Table 5.1: Performance Metrics

The training loss per epoch is depicted in Figure 5.1. The graph shows a significant reduction in loss over the epochs, indicating that the model is learning effectively and converging well.

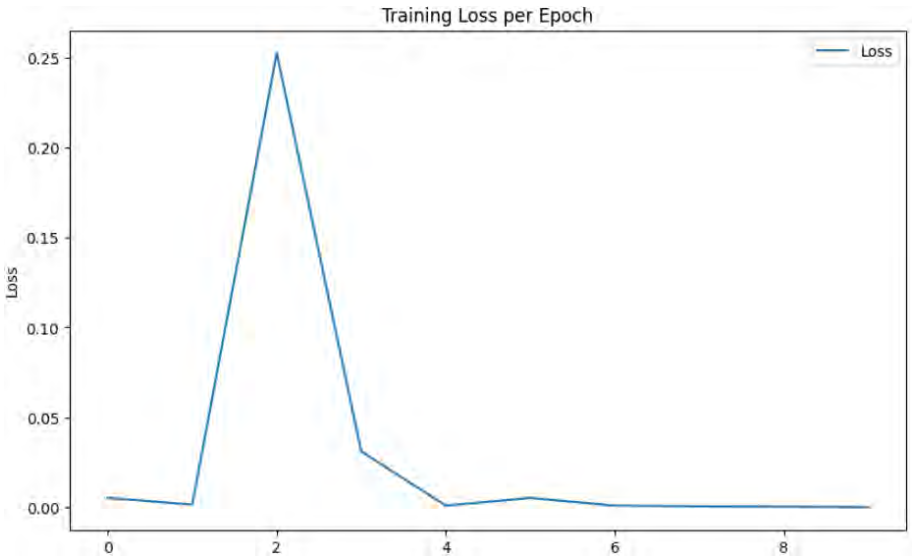

Figure 5.1: Training Loss per Epoch

Figure 5.5 presents a visualization of the model's predictions on sample validation images. Each column shows an anchor image and its corresponding validation image along with the true and predicted labels. This visualization demonstrates the model's capability to accurately classify different types of pneumonia.

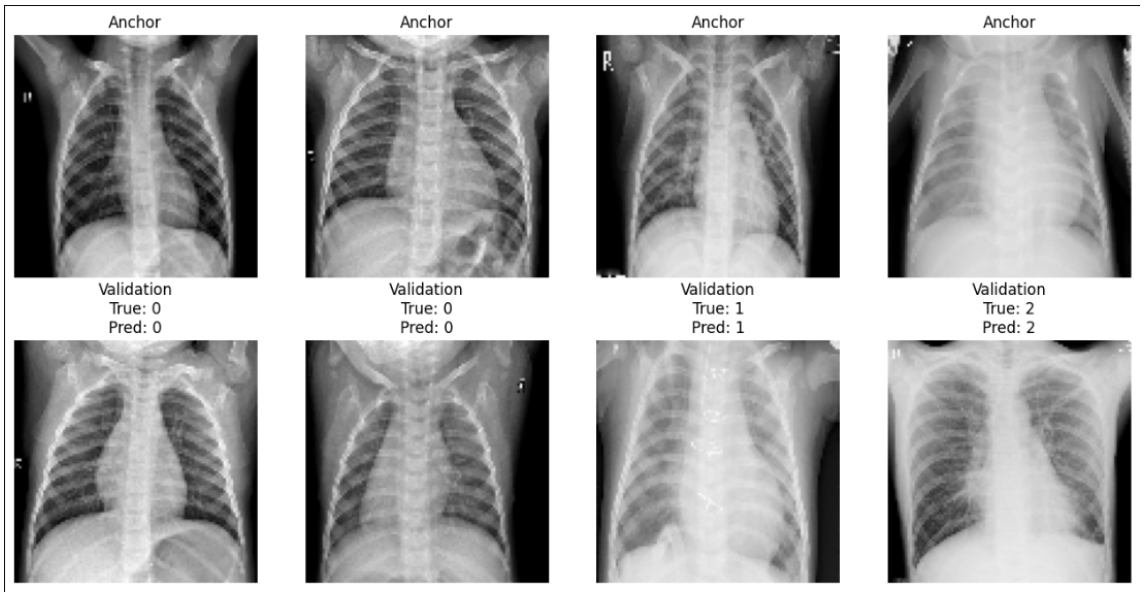

Figure 5.2: Visualization of Predictions: Each column shows an anchor image and its corresponding validation image with the true and predicted labels.

The evaluation metrics, loss graph, and visualization of predictions collectively affirm the proposed model's robustness and reliability in accurately detecting pneumonia from chest X-ray images.

5.2 Distance layer comparison:

The comparative analysis of different distance layers used in the Siamese Neural Network model reveals significant performance variations. Euclidean Distance emerged as the most effective distance layer, demonstrating the lowest loss (0.0092) and the highest recall and precision (0.9902,0.9906), indicating superior accuracy and minimal error. In contrast, Cosine Similarity performed poorly, with both recall and precision at 0, suggesting its unsuitability for this task. Manhattan Distance showed moderate performance with a loss of 0.937 and balanced recall and precision around 0.601, while Hamming Distance had the highest loss (1.2719) and the lowest recall and precision (0.4708 and 0.4711, respectively), reflecting high error rates and unreliable predictions. These results underscore the critical impact of choosing the appropriate distance metric, with Euclidean Distance proving to be the most reliable for accurate similarity comparisons in this context.

Similarity Layers	Loss	Recall	Precision
Cosine Similarity	1.1665	0	0
Eucladian Distance	0.0092	0.9902	0.9906
Manhattan Distance	0.937	0.6014	0.601
Hamming Distance	1.2719	0.4708	0.4711

Figure 5.3: Distance layer comparison

5.2.1 Optimizers:

During the optimization stage of our model's training, we assessed the effectiveness of two distinct optimizers: Stochastic Gradient Descent (SGD) and Adam (Adaptive Moment Estimation). Adam combines the best features of RMSProp and AdaGrad, two more SGD enhancements. For every weight update, SGD keeps track of a single learning rate; Adam calculates adaptive learning rates for every parameter. The Adam optimizer uses the first moment (the mean) and the second moment (the uncentered variance) of the gradients to estimate the learning rate for each weight in the neural network:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{5.1}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{5.2}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{5.3}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{5.4}$$

$$\theta_{t+1} = \theta_t - \frac{\eta \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{5.5}$$

Adam outperforms the other models in our analysis because of its flexible learning rate, which enables greater updates for rare parameters and smaller updates for more common ones. Compared with SGD, which utilizes the same learning rate for all weight updates and necessitates precise calibration, this adaptive approach frequently leads to faster convergence and can handle the sparse gradients on noisy issues more successfully.
To sum up, Adam was a better option for optimizing our Siamese network model due to its versatility, efficiency, and less computing demand while processing sparse data. This required modification of hyper- parameters.

Optimizer	**Accuracy**
Adam (Adaptive Moment Estimation)	0.9906
Stochastic Gradient Descent (SGD)	0.9511

Table 5.2: Comparison of Optimizers

5.3 Confusion matrix

The confusion matrix demonstrates the high classification accuracy of the model across four classes: We have Normal,Bacterial Pneumonia, Viral Pneumonia, and COVID-19 as our four categories of pneumonia. The diagonal elements show the correct classification on the outcome, with 227 of the 227 Normal cases, 218 of the 225 Viral Pneumonia cases, 415 of the 417 Bacterial Pneumonia cases, and 356 of the 359 COVID-19 cases. The number of misclassifications is low and the majority of them is between classes that are in the same category, for example, Bacterial and Viral Pneumonia. This depicts the strength of our model concerning its ability in detecting cases of COVID-19 with almost total precision. Summing up, these outlaws corroborate the high accuracy and stability of our model in diagnosing various types of pneumonia as well as normal conditions, which may be helpful in practical applications.

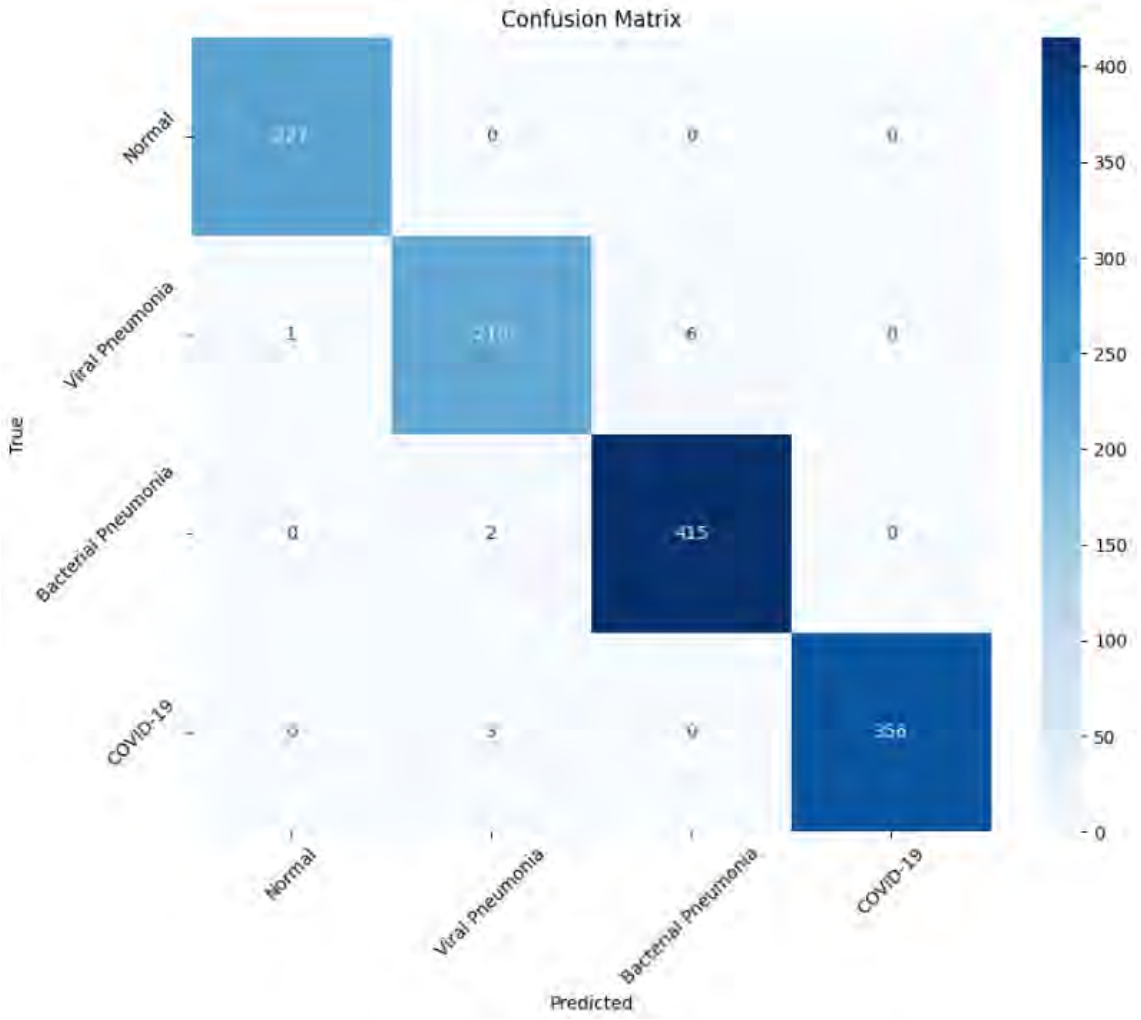

Figure 5.4: Confusion matrix

These results highlight the model's overall high performance, with particularly strong accuracy in identifying Normal and COVID-19 cases.However, the minimal misclassifications which occurred primarily between Viral and Bacterial Pneumonia, suggests areas for further refinement to achieve even higher precision.

5.4 Execution Time Comparison

Model	Average Execution time per epoch
Proposed model for multi-class classification	42s
G Koch's model for binary classification	440s

Figure 5.5: Execution Time

The decrease in average execution times and epochs per mode for multi-class classification when compared to G. Koch's model for binary classification demonstrates the improvement in efficiency. Our proposed model also takes only 42 seconds to finish an epoch, which denotes the model is efficient in handling multi-class problems. This rapid execution time is significant especially in the medical image where timely diagnosis contributes to improved health. The efficiency of our model not only comes with the added benefit of faster processing of large data sets compared to previous method but also makes it practicable to use in real time clinical environment.
Compared to this, G. Koch's model that is designed for binary classification only, takes 440 Sec per epoch which is significantly longer. This longer time may prove to be a drawback when the system is run in a situation where time is critical. Whereas Koch's model may be suitable for the binary classification problems or problems where accuracy is paramount, its longer time to complete one epoch makes it unsuitable for multi-class problems and contexts where real-time image analysis is critical. A significant disparity in the execution time also establishes the relevance of the proposed model in offering rapid, precise, and effective multi-class categorization for health-associated image analysis.

Chapter 6

Conclusion

Hence, in the proposed model of pneumonia, we have documented high accuracy in differentiating between various pneumonia types in chest X-rays. But adding a custom distance layer that has been expedited through CUDA C++ and the utilization of the Euclidean distance measure enhanced the model's accuracy and speed. In addition to achieving higher recall and precision when compared to models such as G Koch's, our models returned results with greater speed, further enhancing our model's real-world usability in the clinical environment. The proficiency with which the model can diagnose pneumonia means that this model could be easily applied in clinical practice, thus giving doctors and other healthcare workers a new powerful tool. Thus, this work emphasizes the necessity of higher techniques in computational method and algorithms to increase the diagnosis accuracy and time efficiency in medical imaging.

6.1 Limitations

However, it is essential to point out that the presented dataset carries several limitations. First of all, the data which is used for the model is quite good but not very diverse and not very large which can lead to low applicability of the model. Thirdly, the number and type of images in the dataset may not be representative of all the variations that one might find in clinical practice, the effect of which when using the model on new images could be suboptimal. It is, nevertheless, crucial to point out that the focus of this investigation is on the construction and assessment of the tailored custom model, and not on the dataset. The contributions of this work are in its design of the new architecture of the custom model and utilization of distance layers.

6.2 Future Work

In future work, we aim to gather large and diverse dataset to analyze the proposed custom model and also try to overcome all those drawbacks which is mentioned in this study. This will require finding datasets that contain the population of patients and pneumonia types that are not present in our data, as well as images with varying degrees of quality. In this way, it has been intended to validate the effectiveness of the presented model as well as improve its reliability and its ability to be adapted for other datasets. Further, the subsequent studies will extend the research through incorporating state-of-the-art data augmentation techniques, CUDA and the transfer learning methodology to enhance the proposed model's performance. By doing so, we hope to succeed in creating a more accurate and stable approach towards pneumonia identification, specifically within medical imaging techniques.

Bibliography

[1] P. Harish and P. J. Narayanan, "Accelerating large graph algorithms on the gpu using cuda," in *High Performance Computing–HiPC 2007: 14th International Conference, Goa, India, December 18-21, 2007. Proceedings 14*, Springer, 2007, pp. 197–208.

[2] M. Garland, S. Le Grand, J. Nickolls, *et al.*, "Parallel computing experiences with cuda," *IEEE micro*, vol. 28, no. 4, pp. 13–27, 2008.

[3] A. Bakhoda, G. L. Yuan, W. W. Fung, H. Wong, and T. M. Aamodt, "Analyzing cuda workloads using a detailed gpu simulator," in *2009 IEEE international symposium on performance analysis of systems and software*, IEEE, 2009, pp. 163–174.

[4] P. Micikevicius, "3d finite difference computation on gpus using cuda," in *Proceedings of 2nd workshop on general purpose processing on graphics processing units*, 2009, pp. 79–84.

[5] R. Farber, *CUDA application design and development*. Elsevier, 2011.

[6] O. Ruuskanen, E. Lahti, L. C. Jennings, and D. R. Murdoch, "Viral pneumonia," *The Lancet*, vol. 377, no. 9773, pp. 1264–1275, 2011.

[7] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, Lille, vol. 2, 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[10] P. Rajpurkar, J. Irvin, K. Zhu, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[11] H. Qassim, A. Verma, and D. Feinzimer, "Compressed residual-vgg16 cnn model for big data places image recognition," in *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, IEEE, 2018, pp. 169–175.

[12] E. Ayan and H. M. Ünver, "Diagnosis of pneumonia from chest x-ray images using deep learning," in *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, Ieee, 2019, pp. 1–5.

[13] C. J. Saul, D. Y. Urey, and C. D. Taktakoglu, "Early diagnosis of pneumonia with deep learning," *arXiv preprint arXiv:1904.00937*, 2019.

[14] O. Stephen, M. Sain, U. J. Maduh, and D.-U. Jeong, "An efficient deep learning approach to pneumonia classification in healthcare," *Journal of healthcare engineering*, vol. 2019, 2019.

[15] T. Gabruseva, D. Poplavskiy, and A. Kalinin, "Deep learning for automatic pneumonia detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 350–351.

[16] Z. Yue, L. Ma, and R. Zhang, "Comparison and validation of deep learning models for the diagnosis of pneumonia," *Computational intelligence and neuroscience*, vol. 2020, 2020.