Analyzing the Security Differential Privacy Provides and the Trade-off between Performance and Privacy in Medical Image Classification

by

Sumaiya Haque
23141039
Mohammad Azim Mehraj
23141050
Mohammad Faiazur Rahman
20101423
Mahmud Abedin
20301366

A thesis report submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

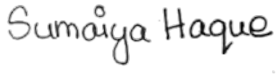
Department of Computer Science and Engineering
Brac University
May 2024

Declaration

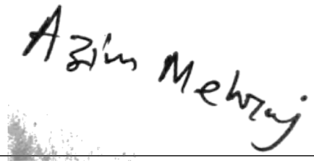
It is hereby declared that

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:



Sumaiya Haque
23141039



Mohammad Azim Mehraj
23141050

Mohammad Faiazur Rahman
20101423

Mahmud Abedin
20301366

Approval

The thesis/project titled "ANALYZING THE SECURITY DIFFERENTIAL PRIVACY PROVIDES AND THE TRADE-OFF BETWEEN PERFORMANCE AND PRIVACY IN MEDICAL IMAGE CLASSIFICATION " submitted by

1. Sumaiya Haque (23141039)
2. Mohammad Azim Mehraj (23141050)
3. Mohammad Faiazur Rahman (20101423)
4. Mahmud Abedin (20301366)

Examining Committee:

Supervisor:
(Member)

Md Tanzim Reza
Lecturer
Department of Computer Science and Engineering
Brac University

Co-supervisor:
(Member)

Rafeed Rahman
lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

One of machine learning's main purposes is to draw out functional and practical information from a set of data while perpetuating the entire privacy by protecting all information. While it might seem a bit hard to maintain, privacy does play a vital role in every sector, and thus, the information must be frequently balanced, especially when extracting sensitive datasets. For instance, medical research or image classification can be considered an important application where patient privacy, as well as the extraction of information, are both of utmost importance [12]. Medical images are details that consist of a patient's private information and are collected from various hospitals, nursing homes, and research institutes. Later on, these images are utilized to infer a patient's physical condition, ultimately leading to an invasion of privacy[10]. In recent years, medical images have become a prominent research and analysis subject, and therefore more and more people are getting affected as their private information is being shared. Thus, in our research, we are going to showcase different ways to defend against information leakage. Differential privacy is considered one of the strongest forms of privacy because we work with privacy-preserving algorithms and learning-based mechanisms. Apart from that, federated learning and image watermarking can also help in preserving privacy. Deep learning techniques that can be utilized to preserve data utilizing Conditional GANs also face particular difficulties when used with medical images. In order to show the optimal method of data preservation, we will attempt to collect a dataset.

Keywords: Privacy preservation, medical research, image classification, medical images, information leakage, differential privacy, privacy-preserving algorithms, mage watermarking, deep learning, data preservation.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

The datasets used in many machine learning systems include private information about people, like their location, contacts, media consumption, and medical history. Serious privacy concerns arise from the possibility that an adversary may identify individuals in the dataset by using what the machine learning algorithm produces. For instance, the discovery of a homophobe in the anonymized Netflix Challenge dataset [8] and the discovery of the health information of the former Massachusetts governor in publicly available anonymized medical databases [1]. This fact led to widespread demand for the creation of data analysis methods that respected individual privacy.

Differential privacy is now an essential element of data analysis that protects user privacy. It gives data scientists and computer scientists a technique to add noise to data in a controlled manner, preventing individual records from being identified but still allowing for the extraction of useful insights [12]. Due to a huge increase in the collection and storage of personal data, including bank account information, census data, and online search history, privacy concerns have increased [10]. This has caused the implementation of data analysis which is privacy preserved. This kind of analysis ensures that the anonymity of an individual is maintained while the data are used in training machine learning models[2][3][14][13].

The privacy and anonymity of the people in the data have been guaranteed through a variety of different methods. For instance, the following is a list of various methods or procedures that have been utilized to protect people's privacy when using machine learning and data science:

- K-anonymity: It is a data privacy technique that makes sure that an individual's data can't be singled out. The way K-anonymity works is that it provides privacy by arranging data points into groups of K data; the groups are formed based on similar non-sensitive features or non-identity-revealing features. This anonymization technique is resistant to attacks like background knowledge attacks[5].

- l-diversity: The idea was put forth by Machanavajjhala et al. in 2007 [5]. The l-diversity scheme was created to overcome various shortcomings in the k-anonymity scheme by increasing the intra-group variability of sensitive data inside the anonymization scheme [6]. Similarity and skewness attacks can be used against it[4].

- T-closeness: It is a different method that builds on the I-diversity methodology. It requires that the distribution of sensitive qualities in each quasi-identifier group be "close" to the distribution of those sensitive attributes throughout the entire original dataset. (i.e., the difference between the two distributions should not exceed a threshold t)[4]. It is possible that achieving a high level of t-closeness may reduce the usefulness of the anonymized dataset [7] by leading to considerable data distortion. The choice of a suitable value for the parameter "t" is difficult to make since there are no set rules for doing so [4].

- Despite having the anonymization techniques mentioned above, differential privacy has become the gold standard for providing privacy in machine learning and data science. The main reason for this is that it allows for strong privacy guarantees. Differential privacy is adaptive to dynamic environments where data continuously evolves, and differential privacy allows its use of quantifiable privacy levels.

1.2 Research Objectives

Medical imaging is significantly important in modern diagnostic procedures. Medical imaging produces a vast volume of data that gives machine learning a great opportunity to train on these data for effortless and quick detection of diseases based on these medical images. Machine learning training on medical images has led to the advancement of medical image classification, therefore significantly reducing the manual time and effort required of medical professionals [27]. These types of early detection of diseases using machine learning can tremendously enhance the quality of treatment patients can receive. Nevertheless, privacy and security problems arise as a result of the inherent characteristics of the data. A substantial obstacle lies in the availability of such annotated medical image data sets, these data sets are confidential and sensitive and are not easily available for openly training machine learning models.

The primary objective of this study centers around the incorporation of differential privacy within the framework of federated learning, specifically for the purpose of categorizing medical photographs. We hypothesize that this framework provides robust privacy guarantees, addressing the ethical, legal, and social implications of data sharing in healthcare applications. Additionally, we explore the balance between privacy and model utility through careful calibration of the privacy budget in DP. This research contributes to the broader discourse on privacy-preserving machine learning and aims to pave the way for secure, privacy-preserving collaborations in medical image analysis.

1.3 Problem Statement

When it comes to training machine learning models on medical images for the diagnosis of diseases or for any other medical purpose, one of the main issues that we have to deal with is the availability of annotated medical datasets and the insecurity of sharing medical datasets or medical images for the sake of patient anonymity.

The US Health Insurance Portability and Accountability Act (HIPAA) and the EU General Data Protection Regulation (GDPR) have strict laws and regulations in place that make it difficult to work with medical images [19]. Medical data and images contain much sensitive information about individuals, like their health condition, age, the treatment they received, their social security number, PIN number, their address, and much more. That's why many medical institutes do not allow their patient's medical data to be used in research and surveys. Some institutes allow their medical data to be used in research facilities and surveys, but they endeavor to anonymize the identities of individuals in the dataset. However, these kinds of anonymized data sets prove to be counterproductive Consider the instance of Governor William Weld's re-identification in an insurance data set with direct identifiers removed [1]. There was another notable incident of re-identification of individuals from a dataset provided by Netflix in 2006. Netflix held a competition with a dataset of around 100 million movie ratings and a dated rating of around 500,000 individuals. The goal of the competition was to check if people could make an algorithm with 10% more accuracy than Netflix; if there was someone with more than 10% better accuracy, then that team would be rewarded with a million dollars. Obviously, the identity of the users in the dataset was anonymized and reduced to just a unique number to make sure which rating belonged to which user. Nonetheless, within two weeks of releasing the dataset, a Ph.D. student, Arvind Narayanan, and his advisor, Vitaly Shmatikov, were able to identify the users by comparing it with another publicly available dataset (IMDB movie rating site) [1]. This shows that data anonymization fails. According to the founders of Cynthia Dwork, one of the inventors of differential privacy, "anonymized data is not". What she means is that data is never anonymized, or it's anonymized so much that it is not data anymore. Our research aims to fulfill the lack of privacy in the medical image dataset, to ensure that using certain methods like differential privacy and federated learning, it's almost impossible to identify individuals in the dataset and to show that these methods work better than anonymization of individuals in the dataset.

Providing privacy for individuals will permit institutions to share their data sets of medical images and non-imagery data sets to be used in surveys and research work. We plan to use differential privacy to ensure the privacy of individuals, to make sure that our output does not trace back to the confidentiality of individuals, and to make sure that our accuracy does not deteriorate much because of the noise added by differential privacy.

Chapter 2

Literaturte Review

2.1 Differential Privacy

The main issue occurs when sensitive and private information is provided for training models for artificial intelligence (AI) algorithms because these algorithms are typically produced through machine learning and require a larger volume of high-quality information. Such leakage of data can cause enormous problems in a patient's life as well as for healthcare providers. As a result, the majority of healthcare institutions have rigorous laws against data sharing, such as the European Union General Data Protection Regulation (GDPR), which assures data security at all costs [32]. A class action lawsuit was filed against Google for breaking UK data protection rules. The data that was breached was related to an AI algorithm that was created to identify patients who were suffering from acute kidney injury. AI models that allow the breaching of sensitive information such as one's personal problem, location, or identity have become a consequential distress for user privacy. Recent studies have also brought out some issues with deploying AI models in the medical sector. Hall et al. stated that patients will not have any trust in the healthcare system if the underlying data has no proper security and is prone to attacks. The necessity of data protection in the era of artificial intelligence was also discussed by Tom et al. The importance of security and innovation has been highlighted by both authors [32].

Medical image classification is meant to assign a medical image or a part of an image to a specific disease or condition [18]. Since many deep learning models, like CNN, are incredibly accurate at detecting cancer from CT scans or identifying skin cancer from dermoscopic pictures, they have become the state-of-the-art method in medical image classification **17**. The effectiveness of these models depends on annotated, sensitive medical training data that is both readily available and of high quality and thus can take a huge amount of time and effort to gather. With the advent of transfer learning, it has become common to train these models on local machines and send the updated weights to the central machine. The machine learning models are trained on several local servers, and then their weights are transmitted back to the main server, where they are incorporated into the model. Despite these advances, the guarantee of data privacy and confidentiality still remains a big chal-

lenge. Lately, there has been much research in federated learning to ensure security. With the help of X-ray images, Zhein Li has presented a federated learning model that can identify COVID-19. His research was notable for employing training loss for each model as the foundation for parameter accumulation weights, which increased efficiency and accuracy. [30]. Similar to this, Jun Luo suggested a method for classification tasks that, utilizing information on the label distribution of clients, strategically modifies the impact of each data sample on the local target during optimization, therefore reducing instability brought on by data heterogeneity. [31]. Another study by Mohammad Adnan showed that a differentially private federated framework can achieve results comparable to conventional training [28]. Differential privacy adds noise to the local models before sending them to the server for integration. This addition of noise has tended to degrade the model's performance, and hence there has to be a tradeoff between privacy and utility. Different strategies have been proposed in many domains, like adjusting the privacy budget allocation based on the model's learning progress [17]. The complexity bound for differential privacy in supervised learning classification was examined in other Chaudhuri and Hsu studies [9]. The differential privacy team at Apple has proposed a scalable and effective local privacy technique [16].

The majority of the models used by researchers to identify COVID-19 instances in hospitals rely on chest X-rays and CT scans. Horry investigated transfer learning for COVID-19 using images from X-rays, ultrasounds, and CT scans [21]. Afshar used the COVID-CAPS capsule structure to study COVID-19 detection in X-ray images [20]. They demonstrated that COVID-CAPS performed better than conventional models. In order to detect COVID-19 utilizing x-rays and CXR pictures, Mukherjee proposed a DNN method (Deep Neural Network) adopted by CNN; their suggested approach outperformed InceptionV3, MobileNet, and ResNet [25]. In order to add controlled noise to the gradient of parameters and clip it during the training of a deep model, Abadi et al. proposed the differentially private stochastic gradient descent (DP-SGD) approach [15]. In their study, Fan et al. examined the performance of four models (MobileNetv2, ResNet18, ResNeXt, and COVID-Net) for COVID-19 detection based on x-ray images. Both in federated and unfederated learning, ResNet18 demonstrated superior performance. Bozkir et al. (2021) offered techniques to safeguard the biometric information that can be detected by our eyes [24]. With the introduction of VR and AR glasses, it has become crucial to protect the biometric information that can be detected by the eyes. To create a segmentation network for CXR pictures, Ziller suggested using a discriminative model trained with DP-SGD (private stochastic gradient descent) [26]. A different researcher, Kossen, proposed leveraging differentially private time-of-flight magnetic resonance angiography (TOF-MRA) images produced by generative adversarial networks (GANs) trained using DP-SGD [29]. It is not immediately clear for DP-SGD the theoretical assurance that images created by GANs trained using DP-SGD meet e-LDP. Other than for medical purposes, a variety of image usage protection techniques are employed. For sharing, retrieving, and feature extraction of images utilizing untrusted sites, many studies have explored cryptography-based solutions. These techniques have the limitation that crypto-based picture sharing expressly trusts the recipients of the data. Sending and receiving data to a variety of people can be difficult [11].

Violation or leakage should also be considered in ways such as privacy from other patients or external staff. In the context of a larger amount of data, security should be set to defend against any malicious attacks on data that can trigger the leakage of sensitive information. Over the past few years, powerful data mining tools across the internet have been used more and more often to exploit sensitive information.

In order to keep data safe and advance research, it is important to maximize the trade-off between privacy and utility. As we examine the results, we can see that utility loss and fairness might vary, which may be related to the range of datasets. Therefore, it is also necessary to use certain techniques to prevent data leakage.

Different kinds of privacy attacks have been tracked over the past few years, such as the re-identification attack proposed by Alam et al. [23]. In such attacks, temporal and spatial information is used separately to identify the exact figure. The breathing rate and heart rate are collected in this framework using a Multi-Modal Siamese Convolutional Neural Network (mmSNN) model in order to re-identify the person.

In medical imaging, anonymization is sometimes used, which requires the removal of relevant DICOM metadata entries such as name, gender, and so on, which helps preserve the main information, which is the illness or disease. Pseudonymization is also used where the real entries are being replaced by artificially generated data, but it is rather a complex process as it is not just data deletion like anonymization but also data manipulation,n which means the actual dataset is being safeguarded somewhere. The de-identification process involves data transfer, and the requirements for this process vary from imaging dataset to imaging dataset [22].

There has been some significant work on medical image classification using differential privacy, but a vast majority of this work was done during COVID time. A large amount of research on the confidentiality of patients was done based on federated learning and differential privacy. We will leverage the prior work as a foundation for acquiring knowledge and implementing differential privacy. Our objective is to discern the optimal trade-off between privacy and performance.

2.2 Neural Network

A computing model called a neural network is modeled after the functioning of organic neural networks seen in the human brain. A crucial element of machine learning (ML) and artificial intelligence (AI) are these models. Layers of linked nodes, or "neurons," make up neural networks. Each layer processes incoming data and sends the result to layers above it. The following are the principal elements and ideas of neural networks:

1. Layers and Neurons:

    - **Neurons:** Resembling the neurons in the human brain, these are the fundamental building blocks of a neural network. Every neuron takes in

information, applies a mathematical function to it, and then sends the result to other neurons.

- **Layers:** The structure of neurons is layered. There are three primary kinds of layers: The initial layer to receive raw input data is known as the input layer. Hidden Layers deal with the data processing of intermediate layers that receive input. The network may learn intricate patterns by having several hidden layers. The output layer is the last layer that generates the network's output.

2. Biases and Weights:

   - **Weights:** The network's parameters that modify incoming data are called weights. As learning progresses, the weight of each neuronal connection changes.
   - **Biases:** Biases are the extra parameters added to the neuronal inputs to improve the model's ability to fit the data.

3. Functions of Activation:
   These mathematical operations are used to add non-linearity to the model by applying them to each neuron's input. The sigmoid, tanh, and ReLU (Rectified Linear Unit) functions are examples of common activation functions.

4. Training:
   Training is the process by which neural networks gain knowledge from data. In order to reduce the difference between the desired and actual outputs (often referred to as the loss or error), the network modifies its weights and biases during training. Backpropagation is a popular training approach that updates the weights and biases by allowing the error to travel backward through the network.

5. Neural Network Types:

   - Feedforward Neural Networks (FNN): The most basic kind in which there are no cycles in the connections between the nodes. From input to output, data flows in a single direction.
   - Convolutional Neural Networks (CNN): Utilizing convolutional layers that apply filters to extract features, CNN specializes in processing structured grid data, such as photographs.
   - Recurrent Neural Networks (RNN): Designed for sequential data, such as text or time series, wherein connections create directed cycles that enable the persistence of information.
   - GANs, or Generative Adversarial Networks: consist of two networks- a discriminator and a generator—that compete with one another to produce representative samples of data.

Due to their superior ability to handle intricate, non-linear relationships in data, neural networks are extensively utilized for a wide range of applications, including voice and picture recognition. Their ability to automatically extract features from

unprocessed data minimizes the requirement for human feature engineering. Convolutional neural networks (CNNs) are used to process images, while recurrent neural networks (RNNs) are used to process sequences. These networks are capable of processing high-dimensional input, including text.

Neural networks are useful for prediction tasks because they can generalize effectively to new data after being trained on big datasets. Additionally, they gain from more recent hardware, such as GPUs and TPUs, which use parallel processing to speed up training and inference. Neural networks have remarkable versatility and adaptability, finding applications across a wide range of domains like robotics, healthcare, and finance, and consistently achieving cutting-edge results. Neural networks are strong tools for quickly tackling a wide range of issues because of their capacity to handle big datasets, learn intricate patterns, and enable end-to-end learning.

Neural networks can learn complicated patterns while integrating privacy-preserving methods, they are highly suited for use in medical picture datasets with differential privacy. These networks are particularly good at spotting minute details in high-dimensional medical images, which are necessary for anomaly identification and precise diagnosis. By including noise in gradients, model parameters, or input data, neural networks can be trained with differential privacy, which prevents the exposure of specific data points. Methods such as Differentially Private Stochastic Gradient Descent (DP-SGD) guarantee that the model keeps performance levels high while maintaining anonymity. Neural networks additionally protect personal data by performing effectively on unseen data thanks to their great generalization capabilities. Furthermore, they can handle large medical imaging collections without experiencing appreciable performance loss thanks to their efficiency and scalability.

Because neural networks can learn complicated patterns while integrating privacy-preserving methods, they are highly suited for use in medical picture datasets with differential privacy. These networks are particularly good at spotting minute details in high-dimensional medical images, which are necessary for anomaly identification and precise diagnosis. By including noise in gradients, model parameters, or input data, neural networks can be trained with differential privacy, which prevents the exposure of specific data points. Methods such as Differentially Private Stochastic Gradient Descent (DP-SGD) guarantee that the model keeps performance levels high while maintaining anonymity. Neural networks additionally protect personal data by performing effectively on unseen data thanks to their great generalization capabilities. Furthermore, they can handle large medical imaging collections without experiencing appreciable performance loss thanks to their efficiency and scalability. Neural networks may adhere to strict data protection laws like HIPAA and GDPR by implementing differentiated privacy, guaranteeing that private patient data is safe while performing a variety of medical imaging activities.

They can also reliably identify intricate patterns necessary for medical diagnostics while guaranteeing patient data is safeguarded, neural networks are perfect for medical image collections with differential privacy. Differential privacy strategies protect personal information while maintaining the model's learning capacity by introducing noise during training. Additionally, neural networks operate reliably

and prevent overfitting by generalizing well to new data. They are an excellent tool for safe and efficient medical image analysis because of their capacity to handle big datasets quickly and their adherence to laws like GDPR and HIPAA.

2.3 Binary Classification

Binary classification is a supervised machine learning task that categorizes data into one of two mutually exclusive classes or categories. The model forecasts a binary result, meaning that it could be 1 or 0, true or false, spam or not spam, etc., or it could be positive or negative. With labelled training data that includes features (variables) and class labels, the binary classification model can identify patterns. It then predicts the class of fresh, unlabeled data using these discovered patterns. A probability that the example belongs to the positive class is produced by the model.

2.3.1 Binary Classification's Applications

- Email Spam Detection: To assist filter out unsolicited emails, binary classification is used to identify emails as spam or not spam.
- Churn Prediction: This tool helps organizations take proactive steps to keep consumers by predicting whether or not they will churn, or depart.
- Conversion Prediction: Businesses can use binary classification to forecast a customer's likelihood of converting, or making a purchase, which helps them tailor their marketing campaigns.
- Medical diagnosis: It is used to categorize people as sick or well, allowing for the early discovery and management of illnesses.
- Financial Fraud Detection: To assist stop financial losses, binary classification is used to identify fraudulent transactions.

2.3.2 Reasons to Use Binary Classification:

By reducing complex difficulties to a simple yes or no answer, binary classification makes the problems easier to understand and analyse.

- Simple to create: Even for people without a lot of machine learning knowledge, binary classification models are comparatively simple to create. Broad Applicability: From marketing to healthcare, binary classification has numerous uses in a variety of industries.
- High Accuracy: When paired with methods like ensembling and hyperparameter optimisation, binary classification models can get a high degree of accuracy.

2.3.3 Use in medical image dataset to work for differential privacy (Binary Classification)

As binary classification algorithms can categorize data into two groups while maintaining the privacy of individual data points, they are frequently employed in medical picture datasets to work for differential privacy. In order to prevent the model from memorizing particular characteristics of individual data points, differential privacy strategies introduce noise into the model parameters during training. This protects sensitive information seen in medical imaging. Healthcare professionals and researchers may efficiently analyze medical images for activities like disease diagnosis, patient monitoring, and treatment planning while protecting patient privacy and confidentiality by using differential privacy in conjunction with binary classification models.

2.4 CNN

Convolutional Neural Networks (CNNs) are a marvel of modern technology in the fields of artificial intelligence and computer vision. They are painstakingly designed to decipher the complex web of visual input. Imagine the network as a collection of linked layers, similar to the visual cortex of a human, with each layer carefully adjusted to extract ever more abstract characteristics from unprocessed input photos. Convolutional layers, which are the brains of the CNN, are composed of tiny, reconfigurable filters that move over the image to pick up subtle patterns like edges, textures, and forms. These layers carefully craft a hierarchical representation of visual features; they are the crafters of vision. Activation functions, like the widely used Rectified Linear Unit (ReLU), give the data movement through the network a nonlinear energy that allows the network to identify increasingly complex patterns with each layer. Then, pooling layers take over, decreasing computing overhead and preventing overfitting by downsampling to extract the essential information from the data. The voyage ends in fully connected layers, where the extracted features come together and meld together to create the fundamental structure of recognition and categorization. Last but not least, the output layer functions as a wise oracle by applying probabilities to the network's predictions, illuminating the identities of objects, scenes, or abnormalities in the pictures. CNNs are essentially the perfect example of how science and art can coexist, combining mathematical precision with artistic vision to reveal the mysteries contained in visual data.

Convolutional Neural Networks (CNNs) have become more well-known and well-liked because of their exceptional ability to solve a broad range of computer vision tasks. Numerous elements have a part in their popularity and praise:

- Hierarchical Feature Learning: From raw input data, CNNs can automatically learn hierarchical representations of features. Similar to how the human visual system organizes visual information hierarchically, they can recognize progressively complex patterns and characteristics at various degrees of abstraction thanks to this hierarchical approach

- Translation Invariance: CNNs are capable of detecting patterns in input images regardless of where such patterns are located. For applications like object identification and recognition, where items may appear at multiple places inside the image, this trait is crucial.

- Parameter Sharing and Sparse Connectivity: CNNs use sparse connection and parameter sharing to drastically cut down on the number of parameters needed in comparison to fully linked networks. Because of their efficiency, CNNs can tackle complicated tasks and vast datasets without becoming computationally prohibitive.

- Pre-Trained Models and Transfer Learning: A solid basis for transfer learning is provided by pre-trained CNN models, which were trained on enormous datasets such as ImageNet. By using smaller, domain-specific datasets to refine these pre-trained models, developers can drastically lower the quantity of labeled data and processing power needed to get excellent performance on novel tasks.

- Scalability and Parallelization: CNN architectures are highly scalable and efficient to train and deploy on contemporary hardware architectures such as GPUs and TPUs because they lend themselves well to parallel processing. Researchers and practitioners can work with increasingly complicated problems and datasets thanks to its scalability.

- Broad Range of Applications: CNNs have proven to be extremely effective in a number of computer vision applications, such as object recognition, picture classification, semantic segmentation, captioning, medical image analysis, and more. Their adaptability and versatility make them essential tools in a variety of fields, such as autonomous vehicles and healthcare.

- State-of-the-Art Performance: CNNs routinely outperform conventional computer vision techniques and even approach human performance in specific tasks, demonstrating state-of-the-art performance on benchmark datasets and real-world applications. CNNs are now at the forefront of computer vision research and industry applications thanks to their exceptional efficacy.

Fundamentally, Convolutional Neural Networks are the cornerstone of modern computer vision and the driving force behind numerous technological advancements due to their remarkable performance across a wide range of tasks, translation invariance, efficiency, scalability, versatility, and capacity to learn hierarchical representations.

Convolutional Neural Networks (CNNs) are a shining example of innovation in the field of medical imaging. They provide a means of protecting patient privacy while addressing the complexities of diagnostic imaging. Imagine a system of linked layers, like a maze of synaptic connections, carefully designed to interpret the minute details included in medical pictures. These networks act as virtuoso interpreters, identifying illness markers, abnormalities, and complex patterns with unmatched precision. Imagine now giving these networks the protection of differential privacy, which hides specific data points but yet permits the network to learn

and draw conclusions with absolute accuracy. Patient privacy is protected without compromising thanks to CNN's artistic ability, as each layer painstakingly pieces together a tapestry of insights while respecting the confidentiality of private medical information. A new age in medical imaging is being ushered in by this harmonious convergence of ethical duty and technological innovation. CNNs have the capacity to illuminate the route toward breakthrough healthcare solutions while maintaining the greatest standards of privacy and confidentiality.

2.5 ResNet-50

A particular kind of convolutional neural network (CNN) architecture known as ResNet-50, or Residual Network with 50 layers, was first described in a 2015 paper by Zhang Xiangyu, Ren Shaoqing, and Sun Jian. It is also known for its creative application of residual blocks, which integrate shortcut connections that "skip over" specific layers in order to solve the vanishing gradient issue and help the network acquire more accurate representations of the input data.

2.5.1 Applications of ResNet-50

- Image Classification: ResNet-50 performs exceptionally well at classifying photos into numerous categories, which qualifies it for use in applications that need precise object recognition.
- Transfer Learning: ResNet-50's pre-trained weights can be utilised as a starting point for fine-tuning on particular datasets, allowing for the effective training of models for a range of applications.
- Object Detection: ResNet-50 has strong feature extraction capabilities and can serve as the foundation network for object detection systems.
- Image Segmentation: ResNet-50's design can be modified for semantic segmentation tasks, in which each pixel in an image has to be given a class name.

2.5.2 Motivation for Using ResNet-50

1. State-of-the-Art Performance: ResNet-50 demonstrates its capacity to learn potent representations of visual input by achieving outstanding performance on difficult benchmarks like ImageNet2.
2. Depth and Efficiency: Training incredibly deep networks with hundreds of layers is made possible by the residual connections in ResNet-50, all while preserving computational efficiency.
3. Versatility: ResNet-50 is a flexible option for a broad range of computer vision applications since it can be readily adjusted to different tasks and datasets.

4. Ease of Training: ResNet-50's residual connections lessen the effect of the vanishing gradient issue, which facilitates the training of deep networks in contrast to conventional architectures.

2.5.3 Use in a medical image dataset to work for differential privacy (Resnet-50)

ResNet-50's exceptional performance, reliable accuracy, and capacity to extract strong features from images make it a useful tool for differential privacy in a medical image collection. ResNet-50's architecture—which consists of deep neural networks and pre-trained weights—allows the model to recognise intricate patterns and structures in medical images, which helps it to classify and analyse various medical conditions while protecting privacy using methods like differential privacy. Furthermore, ResNet-50 has demonstrated its adaptability and efficacy in medical image analysis tasks by being successfully used in a variety of medical imaging tasks, including tumor identification in MRI scans, breast cancer detection, and automatic diagnosis of pulmonary infections in COVID-19 CT images.A supervised machine learning job called binary classification divides data into two classes or categories that are mutually exclusive. The model forecasts a binary result, meaning that it could be 1 or 0, true or false, spam or not spam, etc., or it could be positive or negative.

With labeled training data that includes features (variables) and class labels, the binary classification model can identify patterns. It then predicts the class of fresh unlabeled data using these discovered patterns. A probability that the example belongs to the positive class is produced by the model.

2.6 VGG16

VGG16 was introduced in their 2014 paper "Very Deep Convolutional Networks for Large-Scale Image Recognition." The "16" in VGG16 refers to the 16 weight layers in the network: 13 convolutional layers and 3 fully connected layers. It was developed by Simonyan and Zisserman. The model achieves 92.7% top-5 test accuracy on the ImageNet dataset, which contains 14 million images. It takes an input image of fixed size 224x224 and outputs a vector of 1000 values representing the classification probabilities for each of the 1000 classes in the ILSVRC challenge

2.6.1 Utilizing the VGG16 Model

- Image Classification: VGG16 is frequently used for image classification jobs because of its exceptional accuracy in identifying and classifying objects inside images.
- Feature Extraction: In transfer learning scenarios, when pre-trained models are refined for particular tasks, VGG16 is also employed for feature extraction because of its deep architecture.

- Research and Development: To test novel algorithms, investigate deep learning ideas, and comprehend neural network designs, researchers and developers use VGG16 as a benchmark model.

2.6.2 Motives for Using the VGG16 Model

- High Accuracy: VGG16 has remarkable accuracy rates, which make it appropriate for uses where picture categorization accuracy is essential.
- Versatility: The design of the model is adaptable to a range of tasks, from simple image identification to more intricate applications including visual perception.
- Benchmark success: VGG16 is a dependable benchmark for assessing new models and methods in the field of deep learning because of its success in the ILSVRC competitions and its capacity to surpass earlier models.
- Ease: Notwithstanding its complexity, VGG16's clear-cut and consistent design makes it easier to use and comprehend, making it a viable option for both novices and seasoned deep learning professionals.

2.6.3 Use in a medical image dataset to work for differential privacy (VGG16)

Due to its excellent performance, adaptability, and differential privacy compatibility, VGG16 is a very attractive option for medical picture analysis that protects privacy. Accurate diagnosis and disease detection are made possible by the combination of VGG16 and differential privacy, which also strictly protects patient privacy.

Chapter 3

Dataset and Data Analysis

3.1 Description of the Data

Different approaches to gathering datasets are used in academic research, each with unique benefits and difficulties. Among these techniques is primary data collection, in which scientists collect fresh information from surveys, experiments, or observations. Utilizing pre-existing data from sources like databases, books, or online repositories is known as secondary data collection. Public datasets, like those found on sites like Kaggle, provide a wide range of carefully chosen data that has been gathered from institutional or prior research sources. Furthermore, data can be gathered via crowdsourcing, case studies, simulations, and longitudinal research.

We chose to use a Kaggle dataset for my research for a number of good reasons. Kaggle saves a lot of time and money by giving users access to readily available, well-organized datasets, which would otherwise need to be collected through primary data collection.The dataset comprises subfolders for each image category (Pneumonia/Normal) and is arranged into three folders (train, test, and val). There are two categories (Pneumonia/Normal) and 5,863 X-ray images (JPEG). Anterior-posterior chest X-ray images were chosen from retrospective cohorts of pediatric patients from Guangzhou Women and Children's Medical Center, Guangzhou, aged one to five. Every chest X-ray image was taken as a standard clinical procedure for the patients. All chest radiographs were first screened for quality control by eliminating any low quality or unreadable scans before being subjected to the analysis of chest x-ray images. Before the images' diagnoses could be used to train the AI system, they were evaluated by two board-certified medical professionals.A third expert verified the evaluation set to make sure there were no grading errors.

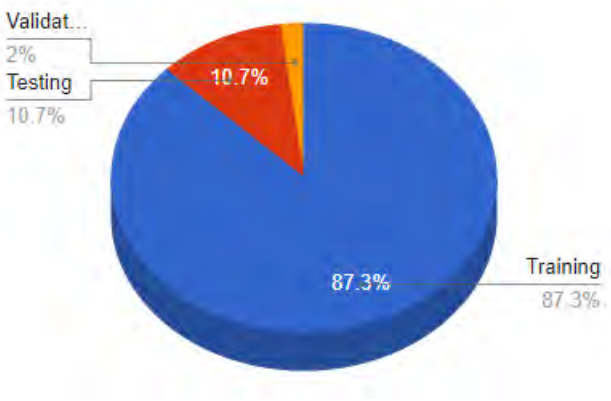


Figure 3.1: train-test pi-chart

We split our dataset into 3 sets training, validation, and testing. The dataset comprises a total of 5,863 X-ray images, which are divided into three subsets: training, testing, and validation. The training set consists of 5,216 images, accounting for approximately 86.95% of the total dataset. The testing set includes 624 images, making up about 10.64% of the dataset. Finally, the validation set contains 16 images, representing around 2.27% of the total.

It would have been extremely difficult for me to gather such a dataset on my own; I would have needed to work closely with medical institutions, get ethical approvals, and make sure that patient privacy and data security were protected. Moreover, board-certified medical professionals would need to be involved due to the expertise needed to accurately grade the radiographs, which would add complexity and cost. I was able to take advantage of a resource that had already been painstakingly selected and verified by professionals by using an existing dataset from Kaggle. This allowed me to concentrate on the analysis and application of the data in order to effectively build and train the AI system.

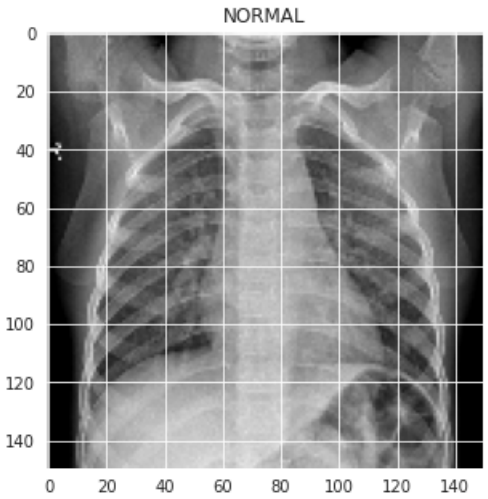


Figure 3.2: Normal

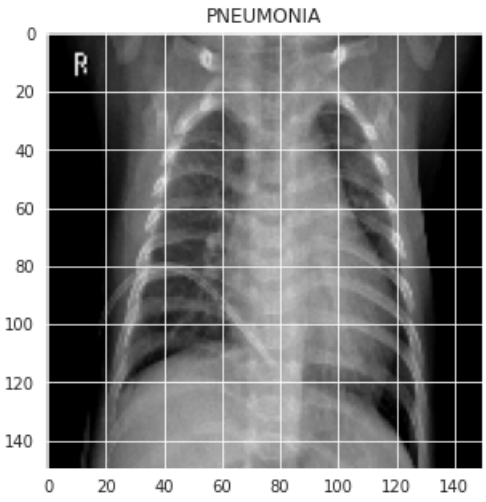


Figure 3.3: Pneumonia

3.2 Data Analysis and Data Pre-processing

Preprocessing is the set of operations performed on unprocessed data to get it ready for analysis or model training. Preprocessing, when applied to models that use images, entails a variety of adjustments and modifications to make sure the images are of a quality and format that the model can use to learn from.In order to prepare the image dataset for training the AI model, the research carefully carried out data preprocessing and augmentation, with the goal of improving the model's performance and making sure it generalizes well to new data. Preprocessing was essential to converting unprocessed image data into a format that could be used by models, which greatly increased the efficacy and efficiency of the training procedure. The different preprocessing methods that were used are described in detail in this document. These methods include formatting, noise reduction, size standardization, data augmentation, normalization, and quality enhancement.

3.2.1 Normalization

A crucial step in the preprocessing process was normalization, which involved rescaling the image's pixel values to a standard range, usually between 0 and 1. Neural networks are extremely sensitive to the amount of input data, so this step is essential. We achieved consistency throughout the dataset by normalizing the pixel values, which is essential for reliable and efficient model training. Better overall performance is achieved during the training phase when faster convergence is facilitated by normalized data. The pixel values were adjusted during the rescaling process to ensure that the image intensity values were uniformly distributed within the given range. This conversion guarantees that the model learns from features rather than random intensity values and helps to mitigate problems caused by the images' fluctuating lighting conditions.

3.2.2 Data augmentation

Data augmentation techniques were widely used to address the imbalance in the dataset and prevent overfitting. When a model performs remarkably well on training data but is unable to generalize to new, unseen data, this is known as overfitting. By artificially increasing the diversity of the training dataset and simulating various real-world scenarios that the model might encounter, data augmentation helps mitigate this problem. In order to help the model become invariant to the orientation of the chest X-rays and guarantee that it could recognize features regardless of how the image was rotated, augmentation techniques were employed, including randomly rotating images up to 30 degrees.Furthermore, the images underwent random horizontal flips, which strengthened the model's resistance to variations in the X-rays' viewing directions. To ensure that the model could handle variations in image positioning, images were also randomly shifted up to 10% of their total width and height, respectively, in both directions. This allowed for the simulation of small positional changes. Another method was zooming, which involved randomly enlarging images by up to 20% in order to aid the model's ability to identify features at various scales. The effect of tilting the image along one axis was simulated using shear transformations, which improved the model's capacity to generalize from different viewpoints. The ImageDataGenerator class from Keras was used to implement these augmentation techniques because it offered a complete framework for applying these transformations dynamically during training. The model's robustness was greatly enhanced by the use of data augmentation, allowing it to function well under a variety of real-world variations.

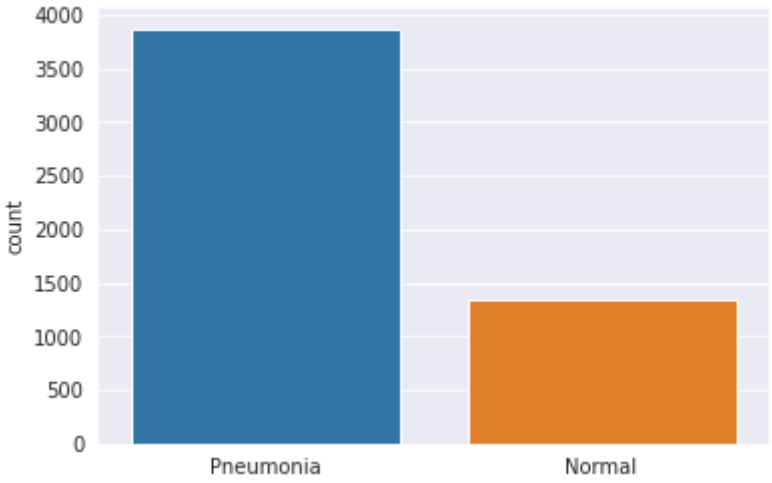


Figure 3.4: augmented bar-chart

3.2.3 Size Standardization

Another crucial step in the preprocessing process was size standardization, which involved resizing the images to fit the model's specified dimensions. This guaranteed consistency in the size of the input, which is necessary for batch processing and effective computing. Similar to the one used in this study, convolutional neural

networks (CNNs) anticipate input images of a fixed size. Ensuring that all images have a uniform size allows for batch processing, utilizing the hardware's full computational capacity and enhancing training effectiveness.To avoid distortion, each image's dimensions were changed during the resizing process while keeping its aspect ratio intact. This step was essential to preserving the integrity of the data by making sure that the resizing process did not change the features in the images. We made sure the model could effectively learn and process features across all images without being hampered by dimensional inconsistencies by standardizing the image sizes.

3.2.4 Formatting

Preprocessing images into a format that the model expected was another crucial step. This involved normalizing color values in accordance with the model architecture's requirements or converting RGB color channels to grayscale. Certain models require specific preprocessing functions in order to format the data correctly, such as the ResNet50 model used in this study. For example, the input data was preprocessed using the ResNet50 model's preprocess_input function before being fed into the model. This function performed tasks such as: modifying the picture dimensions to conform to the ResNet50-expected input size, scaling the pixel values to a range that corresponds to the model's pre-trained weights, and converting labels for use in classification tasks into a categorical format that the model can understand.

In order to prepare the input data for the model, the preprocessing steps included careful normalization to scale the pixel values, extensive data augmentation to increase the diversity of the training data, and the application of a specialized preprocessing function. Together, these actions made sure the model could handle variability in the training set, improve generalization to new data, and improve accuracy when it came to classifying pediatric chest X-ray images into the Normal and Pneumonia categories. Rather than gathering data on its own, the clever use of the Kaggle dataset made use of pre-existing, high-quality, and well-validated data, which resulted in significant time and resource savings as well as reliable and robust model training.

Chapter 4

Methodology, Architectures, and Model Implementations

4.1 System Architecture

4.1.1 CNN Architecture (Without Differential Privacy)

The CNN model is designed for image classification, starting with an input layer for 150x150 pixel grayscale images. The initial Conv2D layer uses 32 filters with a (3,3) kernel size, 'relu' activation, strides of 1, and 'same' padding. This is followed by BatchNormalization for stability and a MaxPooling2D layer with a (2,2) pool size and strides of 2 for downsampling. A second Conv2D layer with 64 filters, 'relu' activation, and similar configuration is added, followed by a Dropout layer with a 0.1 rate to prevent overfitting, another BatchNormalization, and a MaxPooling2D layer. A third Conv2D layer with 64 filters and another MaxPooling2D layer further reduces the feature map dimensions. Typically, a Flatten layer is used to convert a 2D vector to a 1D vector, followed by Dense layers for classification. The model used, is compiled with the 'rmsprop' optimizer and 'binary_crossentropy' loss function, using 'accuracy' as the performance metric.

4.1.2 CNN System Architecture (With Differential Privacy)

To safeguard specific data points during training, the system architecture integrates differential privacy into a convolutional neural network (CNN). TensorFlow Privacy modules are used to accomplish this, notably, the DPKerasSGDOptimizer, which adds noise to the gradients. Important variables that balance privacy and model accuracy are num_microbatches, noise_multiplier, and l2_norm_clip. The architecture consists of layers for feature extraction (Conv2D), training stability (BatchNormalization) and downsampling (MaxPooling2D), overfitting prevention (Dropsout layers), and classification (Dense layers). When combined with the 'binary_crossentropy' loss function and assessed by accuracy measures, this CNN model complies with strict privacy rules and performs an efficient classification of

images, making it appropriate for use in sensitive data applications.

Differential privacy is integrated into the CNN model architecture to safeguard specific data points while they are being trained. An input layer with 150x150 pixel grayscale photos is the first layer it uses. This initial convolutional layer has 32 filters with a (3,3) kernel size and 'relu' activation. To stabilize and speed up training, a Batch Normalization layer comes next. The features are downsampled using a MaxPooling layer that has a (2,2) pool size and padding='same'. A second convolutional layer with 64 filters, a second Batch Normalization layer, a Dropout layer with a 10% rate to avoid overfitting, and a second MaxPooling layer are added after the first. MaxPooling, Batch Normalization, and a third convolutional layer follow this pattern. The dense classification layer receives the output after it has been flattened into a one-dimensional array. In order to guarantee privacy during training, the model is constructed using an optimizer created especially for differential privacy, which includes parameters like l2_norm_clip, noise_multiplier, num_microbatches, and learning_rate. With the integration of differential privacy techniques with convolutional, pooling, normalizing, dropout, and dense layers, this architecture aims to preserve privacy guarantees and efficiently carry out image classification tasks.

4.1.3 Resnet System Architecture and methodology (Without Differential Privacy)

First, a ResNet50 model pretrained on the ImageNet dataset is used in the procedure. Using features acquired from a large and varied image dataset, this pretrained model functions as the foundational model. The ResNet50 basic model's layers remain frozen, which means that during training, their weights are not changed. This method enables the use of ResNet50's powerful feature extraction capabilities without changing the learnt weights, which can expedite training and enhance task performance. Custom fully connected (FC) layers are added to the frozen ResNet50 basis to customize the model for the particular classification assignment. Three dense layers, each containing 256, 512, and 1024 neurons, make up these layers. The ReLU activation function is used by each dense layer to introduce non-linearity, which aids in the model's ability to recognize intricate patterns in the input. To avoid overfitting, each dense layer is followed by a dropout layer with a rate of 0.5. Dropout helps to regularize the model by randomly changing a portion of the input units to zero during training. A softmax layer with two neurons, representing the two classes (pneumonia and normal), makes up the last layer of the custom architecture. The model may produce a probabilistic forecast for each input image by using the softmax activation function, which generates a probability distribution over the classes. The stochastic gradient descent (SGD) optimizer is used to construct the model. SGD is selected because of how well it handles high-dimensional, large-scale data. To enhance convergence, SGD's learning rate and momentum parameters can be changed. Identifying between two classes is the classification task, hence the binary cross-entropy loss function is employed. In order to direct the optimization process during training, this loss function calculates the difference between the actual class labels and the anticipated probabilities. Multiple callbacks are used to

keep an eye on the training process and store the best-performing model. When the tracked metric (usually accuracy) gets better, the ModelCheckpoint callback saves the model weights to a given file directory. This guarantees the preservation of the optimal iteration of the model. TensorBoard may be used to visualize a variety of training metrics, including loss and accuracy, which are logged using the callback function. This image aids in understanding training dynamics and in recognizing problems like as under- or overfitting. Over the course of 12 epochs of training, the model gains the ability to categorize input photos by modifying its weights in accordance with the training set. Batches of augmented images from the training and validation sets are provided by the *train_generator* and *val_generator*, respectively. Throughout the epochs, metrics like training accuracy and loss are monitored. These measures are plotted after training to show the model's evolution in performance.

Ultimately, a classification report produced by contrasting the predicted labels with the test set's true labels is used to assess the model's performance. This report provides a thorough performance evaluation for each class by including precision, recall, f1-score, and support. To see how many predictions are right and wrong for each class, a confusion matrix may also be created. This provides further information on the model's advantages and disadvantages in terms of class distinction.

4.1.4 Resnet System Architecture and Methodology (With Differential Privacy)

The model starts with a ResNet50 base that has been pre-trained on ImageNet using Keras and TensorFlow. All of its layers are frozen to preserve the features that have been learned. Custom fully connected (FC) layers with 1024, 512, and 256 neurons are layered on top of this base and, in order to prevent overfitting, each layer is followed by a dropout layer with a 0.5 dropout rate. A softmax layer for splitting the data into two classes is the last layer. The ImageDataGenerator is used for data augmentation, doing transformations like rotation, flips (horizontally and vertically), and shifts to improve the training set. The use of the tensorflow_privacy package to incorporate differential privacy is the setup's primary difference. The gradients' privacy is ensured during training by the optimizer, DPKerasSGDOptimizer, which is set up with an L2 norm clip of 1.5, a noise multiplier of 0.1, and 8 micro batches. To adhere to differential privacy rules, the reduction technique is set to 'none' and the loss function is binary cross-entropy. The model is trained through 12 epochs, and callbacks such as TensorBoard for logging metrics and ModelCheckpoint for preserving the optimal model weights are used to track performance. The learning rate is also modified based on validation correctness using a ReduceLROnPlateau callback. Following training, a confusion matrix, accuracy and loss plots, and a classification report are used to assess the model's performance and provide a thorough examination of its prediction skills while maintaining privacy-preserving training procedures.

4.1.5 VGG16 System Architecture (without Differential Privacy)

A well-known deep convolutional neural network is the VGG16 model architecture. The processing of photos begins with an input layer of shape (None, 224, 224, 3). Several convolutional layers, including *block*1_*conv*1 and *block*1_*conv*2, are included in the model; they are all tasked with identifying different features in the input images. Further blocks, such as *block*3_*conv*1, *block*3_*conv*2, *block*3_*conv*3, *block*4_*conv*1, *block*4_*conv*2, *block*4_*conv*3, *block*5_*conv*1, *block*5_*conv*2, and *block*5_*conv*3, follow these convolutional layers and extract even more complex patterns from the photos. Together with dense layers like Flatten and Dense for classification, the architecture also incorporates pooling layers to minimize spatial dimensions. The VGG16 model is adept at identifying a wide variety of visual patterns because it was pre-trained on our dataset. This architecture is painstakingly made to recognize and classify images precisely by capturing hierarchical elements in photos.

4.1.6 VGG16 System Architecture (with Differential Privacy)

In order to improve privacy protection without sacrificing model performance, the system architecture is altered when differential privacy is implemented on the VGG16 model. An input layer shaped like (None, 224, 224, 3) is part of the architecture for processing images. It uses convolutional layers to extract features from the input images, like Conv2D, with different filter sizes and activation algorithms. The activations of the preceding layer in each batch are normalized through the use of batch normalization layers. Downsampling and decreasing spatial dimensions are accomplished via MaxPooling2D layers, while overfitting is avoided by including Dropout layers, which randomly set a portion of the input units to zero during training. Moreover, Dense layers are integrated into the model for classification applications.

The DPKerasSGDOptimizer, which ensures privacy-preserving training by incorporating parameters like *num_microbatches*, *noise_multiplier*, and *l*2_*norm_clip*, implements differential privacy. During optimization, the learning rate is changed to regulate the step size. Sensitive data in the training set is safeguarded by incorporating differential privacy techniques into the VGG16 model architecture, which improves the model's resilience and privacy assurances in picture classification tasks.

4.2 Workflow

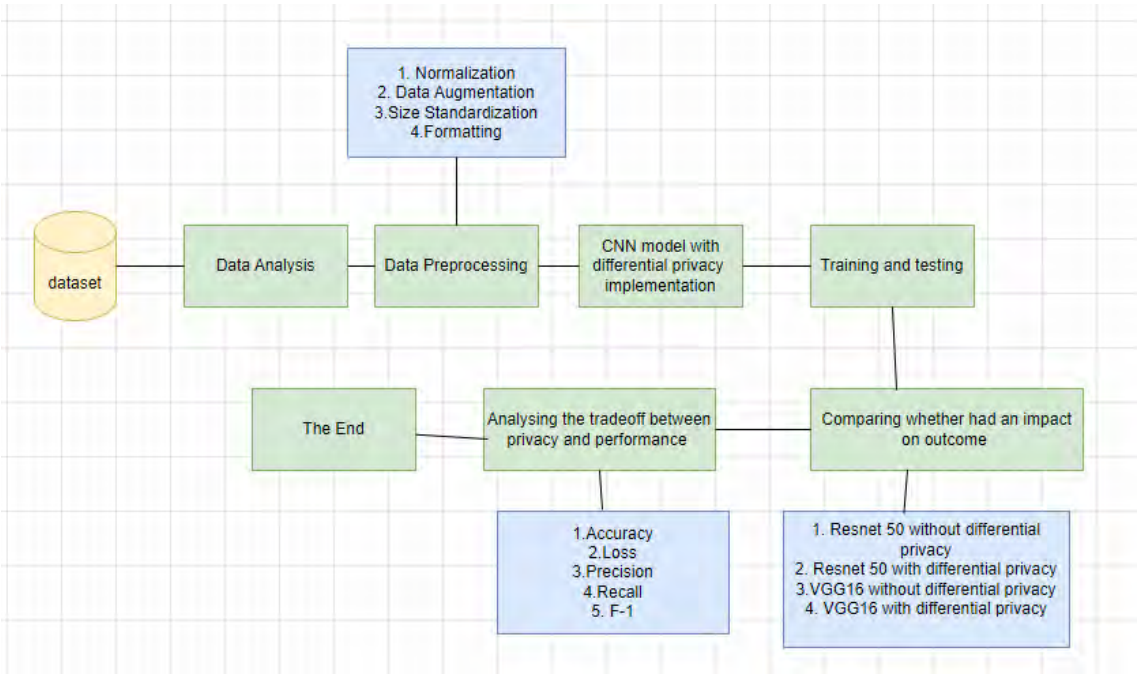


Figure 4.1: Workflow

4.3 Experimental Setup

4.3.1 CNN Setup (Without Differential Privacy)

A number of essential elements are involved in the CNN model's experimental configuration. Convolutional layers with certain settings for filter size, activation functions, and padding make up the model architecture. Normalization and downsampling are then accomplished using BatchNormalization and MaxPooling2D layers. Dense layers are used for classification, while dropout layers are inserted to minimize overfitting. Compiling the model involves using the optimizer 'rmsprop' and the loss function $binary_crossentropy$, with the metric $accuracy$being monitored. Furthermore, the implementation of a ReduceLROnPlateau callback modifies the learning rate in accordance with the validation accuracy. A predetermined batch size, number of epochs, and number of steps are used to train the model. Data preprocessing techniques are used, such as ImageDataGenerator-assisted image augmentation and normalization. The experimental setup also defines the following parameters: dropout, NUM_EPOCHS, $STEPS\_PER\_EPOCH$, $class\_list$, $BATCH\_SIZE$, dropout, and FC_LAYERS. The model's performance is assessed using the $accuracy\_score$ function on the test set of data.

4.3.2 CNN Setup (With Differential Privacy)

To guarantee differential privacy, the CNN model's experimental setup involves setting up variables like *l2_norm_clip*, *noise_multiplier*, *num_microbatches*, and *learning_rate*. The DPKerasSGDOptimizer from TensorFlow Privacy is used to construct the model, and it adds noise to gradients for privacy. Convolutional, pooling, normalizing, dropout, and dense layers make up the architecture, which is intended to protect privacy while efficiently identifying images.

4.3.3 Resnet Setup(Without Differential Privacy)

The experimental setup for the RESNET50 model without any differential privacy entails loading data from specified directories for training and testing pretrained models related to chest X-ray pictures. The document establishes the model architecture using ResNet50 with pretrained weights from the dataset, which excludes the top layer for transfer learning and imports the required libraries, and sets the picture size to 150x150 pixels. The model is then assembled, and trained on the training set of data, and its correctness is assessed on the test set of data.

4.3.4 Resnet Setup(With Differential Privacy)

The approach starts with defining the folders for training and testing data pertaining to chest X-ray pictures in the experimental configuration for the RESNET50 model with the incorporation of differential privacy. After that, the model architecture is created using ResNet50, omitting the top layer for transfer learning, using pretrained weights from ImageNet. To ensure privacy during training, differentially private optimizer DPKerasSGDOptimizer is used in conjunction with parameters like *l2_norm_clip*, *noise_multiplier*, *num_microbatches*, *learning_rate*, epochs, and *batch_size*. Compiling, training, and fitting the model to the data with callbacks for learning rate reduction are done using the designated data directories. By including differential privacy concepts in the RESNET50 model's training process, this arrangement improves privacy protection while the model is learning.

4.3.5 VGG Setup(Without Differential Privacy)

Designing data directories for the training, testing, and validation datasets is part of the experimental setup. Using ImageDataGenerator, image data is preprocessed with validation splitting and rescaling. For frozen layer feature extraction, the pretrained VGG16 model is employed. Additional Dense and Flatten layers are added to the model specifically for categorization. The SGD optimizer and categorical cross-entropy loss are used in its compilation. The image datasets with the designated epochs and batch sizes are used to train the model. Training progress and performance are shown using Matplotlib, and accuracy and loss metrics are used to assess its performance. This configuration shows the VGG16 model's strengths

in image recognition tasks by allowing it to learn and classify chest X-ray pictures efficiently.

4.3.6 VGG Setup(With Differential Privacy)

When differential privacy is applied to the VGG16 model, the experimental setup involves several key components. These include defining parameters such as $l2\_norm\_clip$, $noise\_multiplier$, $num\_microbatches$, learning rate, epochs, and batch size to facilitate privacy-preserving training. The architecture incorporates the DPKerasSGDOptimizer for differentially private optimization, ensuring that the model's weights are updated in a privacy-preserving manner. The model architecture remains consistent with the standard VGG16 model, featuring layers such as Conv2D, BatchNormalization, MaxPooling2D, and Dense layers for feature extraction and classification. Dropout layers are included to prevent overfitting. The differential privacy mechanisms are integrated into the optimizer to introduce noise to the gradients during training, thereby enhancing privacy protection. By adjusting these parameters and incorporating differential privacy techniques, the VGG16 model maintains its classification capabilities while providing privacy guarantees for sensitive data in image classification tasks.

4.4 Model Implementation

4.4.1 CNN Model Implementation (Without Differential Privacy)

The implemented CNN model, which does not include any differential privacy, is organized as a sequential neural network. It starts with a Conv2D layer that uses the same padding and the relu activation function to create 32 filters, each of which has a size of (3,3). This first layer is designed to handle input photos with dimensions of (150,150,1). After the Conv2D layer, the features are standardized using BatchNormalization, and the data is efficiently downsampled using a MaxPooling2D layer with a pool size of (2,2).

The filter size, activation function, and padding are then retained when more Conv2D layers with 64 and 128 filters are added. To avoid overfitting, dropout layers with dropout rates of 0.1 and 0.2 are positioned carefully. Interspersed batch-normalization layers improve the model's performance even more. In order to restructure the data into a manner that is appropriate for the ensuing Dense layers, the model design also has flattened layers.

Units with values of 128 and 256 make up the Dense layers, which use the relu activation function. For regularisation, an additional Dropout layer with a rate of 0.2 is added. Designed for binary classification problems, the final Dense layer has one unit and a sigmoid activation function.

The *binary_crossentropy* loss function and the *rmsprop* optimizer are used for model compilation, and the 'accuracy' measure is used to assess the model's performance. To further optimize the model's learning process, a ReduceLROnPlateau callback is defined to dynamically modify the learning rate during training based on the validation accuracy.

The model is painstakingly built to effectively handle image data, extract pertinent features using convolutional layers, and produce precise binary classifications. It also incorporates necessary methods such as regularisation, normalization, and adaptive learning rate adjustment for improved performance.

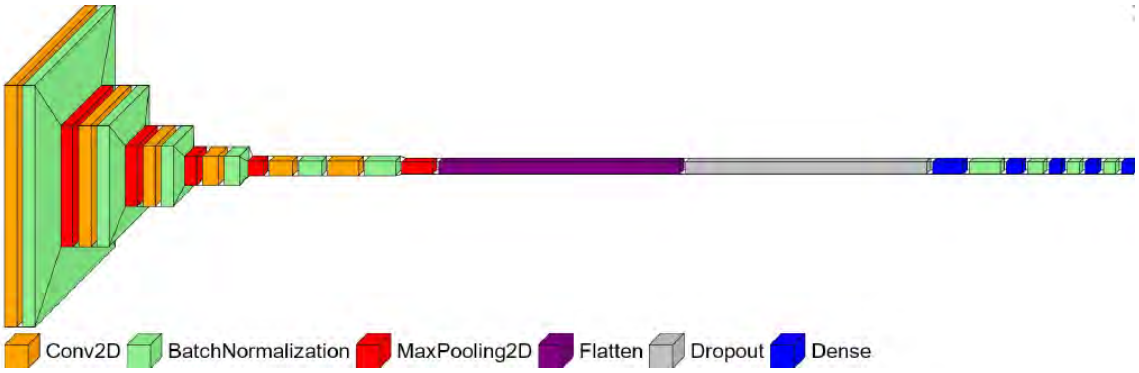


Figure 4.2: CNN model architecture without differential privacy

4.4.2 CNN Model Implementation (With Differential Privacy)

To improve data protection during training, the CNN model implementation with differential privacy has differential privacy parameters built into the model architecture. To guarantee privacy-preserving training, the model specifically makes use of differential privacy approaches like *num_microbatches*, *noise_multiplier*, and *l2_norm_clip*.

Starting with Conv2D layers, the model also contains BatchNormalization, MaxPooling2D, and Dropout layers for downsampling and feature extraction. Differential privacy parameters are introduced into the design to augment privacy guarantees by adding noise and perturbations to the gradients.

The highest Euclidean (L2) norm that the gradients can have is specified by the differential privacy parameter *l2_norm_clip*. This parameter limits the influence of any one training data point on the training process overall by controlling the model's sensitivity to individual training data points. The model contributes to privacy protection by guaranteeing that the gradients used to update the model parameters are constrained by setting an appropriate value for *l2_norm_clip*.

The amount of noise supplied to the gradients during training is specified by the *noise_multiplier* parameter. While more privacy protection is achieved with a greater *noise_multiplier* number, the accuracy of the model may suffer. In order to attain the required degree of privacy without sacrificing the model's functionality,

noise_multiplier must be balanced.

Furthermore, each training data batch is divided into microbatches by the *num_microbatches* option, enabling the independent addition of noise to each microbatch. This separation lessens the impact of individual data points inside a batch, thereby improving privacy guarantees even further.

4.4.3 Resnet Model Implementation (Without Differential Privacy)

The first step in implementing the RESNET50 model without differential privacy is importing the required libraries and modules, which include Matplotlib, TensorFlow, and Keras. The top layer is left out for transfer learning and the model architecture is constructed using ResNet50 with pretrained weights from ImageNet. (HEIGHT, WIDTH, 3) is the input shape specification, with HEIGHT and WIDTH set to 150.

For model evaluation during training, an optimizer (SGD), a loss function ("*binary_crossentropy*"), and metrics ("*accuracy*") are compiled with the model. In order to
visualize and track the training process, a TensorBoard callback is configured, and a checkpoint is made to store the model weights.

The designated data directories for training and testing are then used to train the model. Using predetermined parameters like batch size, epoch count, and step count per epoch, the model is fitted to the training set of data during the training phase. A summary of the model architecture is also provided by printing the model summary. The overall goal of the RESNET50 model implementation without differential privacy is to emphasize model performance without privacy-preserving methods by developing, assembling, and training the model for image classification tasks utilizing the given architecture and training inputs.

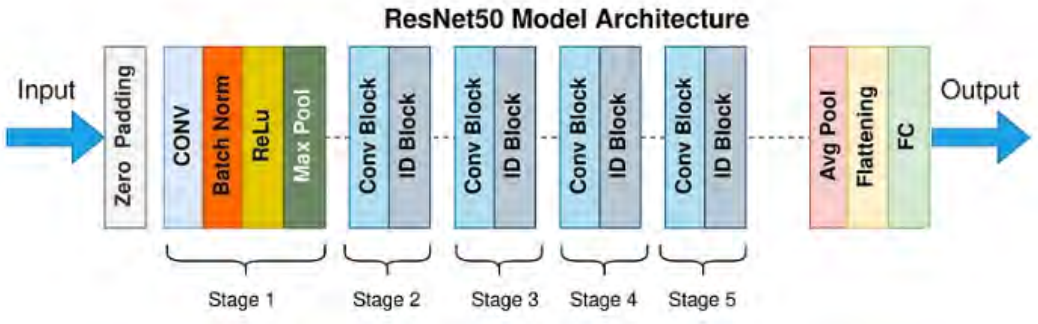


Figure 4.3: Resnet50 model architecture

4.4.4 Resnet Model Implementation (With Differential Privacy)

Building the model architecture using ResNet50 and pre-trained weights from the dataset—aside from the top layer for transfer learning—is the first step in implementing the RESNET50 model with differential privacy. In order to integrate differential privacy, the document defines the input shape and sets many parameters, including *l2_norm_clip*, *noise_multiplier*, *num_microbatches*, *learning_rate*, epochs, and *batch_size*.

Additionally, DPKerasSGDOptimizer, a differentially private optimizer, is chosen to oversee the training procedure using privacy-preserving techniques. With metrics configured to monitor correctness during training, the model is assembled using the specified optimizer and loss function. The document also outlines how to create a checkpoint file to save the model weights for later usage during training.

After that, the model is trained by fitting it to the data using predetermined parameters utilizing the designated data folders for training and testing. The RESNET50 model is trained using this model implementation, which successfully incorporates differential privacy approaches, protecting privacy without compromising model performance.

4.4.5 VGG Model Implementation (With Differential Privacy)

One of the popular deep convolutional neural networks (CNN) that has been widely used for image identification applications is the VGG16 model architecture. The design was developed by the University of Oxford's Visual Geometry Group (VGG), and its uniform structure and simplicity have made it popular in the computer vision sector. The input layer of the VGG16 model is initially created to handle images of the shape (None, 224, 224, 3), where 224 x 224 stands for the spatial dimensions (height and breadth) and 3 for the three RGB color channels. Different numbers of images can be used for training and inference since the None dimension supports a configurable batch size.

Multiple convolutional layers, which are in charge of identifying different features in the input images, make up the core of the VGG16 model. To capture fine-grained patterns, these layers use small receptive fields of size 3x3 (with stride 1 and padding 1). The network is made up of several convolutional blocks, each of which has ReLU (Rectified Linear Unit) activation functions after a number of convolutional layers. The first block consists of two convolutional layers, called *block*1_*conv*1 and *block*1_*conv*2, and a max pooling layer that takes the maximum value from non-overlapping 2x2 regions to minimize the spatial dimensions. Blocks after that follow this pattern while adding further convolutional layers. The feature extraction procedure is further improved by the addition of three convolutional layers to each of the third, fourth, and fifth blocks. Max pooling layers are used to minimize the spatial dimensions of the feature maps after each set of convolutional layers.

This helps to lower the computational effort and avoid overfitting. By choosing the largest value from 2x2 regions, these pooling layers effectively down-sample the feature maps without sacrificing the most important information.

This layer uses a softmax activation function to generate probability distributions over the classes, enabling the model to predict the class of the input image with high accuracy.

The design shifts to a set of fully connected (dense) layers that conduct the classification operation after the final pooling layer. To prepare the data for fully connected layers, the flattening layer is the initial stage, converting the 2D feature maps into a 1D vector. Two completely linked layers with 4096 neurons each are then included in the network, both of which are followed by ReLU activation functions. These layers are made to create a high-level representation of the input images by combining the features that were learned in the convolutional layers. With 1000 neurons in its last layer, the VGG16 model represents the 1000 classes in the dataset. This layer generates probability distributions over the classes using a softmax activation function, which allows the model to accurately predict the class of the input image. VGG16 can capture features at several levels of abstraction because of its hierarchical structure. While higher-level features like forms and object pieces are captured by deeper layers, early layers concentrate on low-level features like edges and textures. Accurate image identification and classification depend on this hierarchical feature extraction.

The first step in implementing the VGG16 model is to build up the basic model. To ensure that the model has learnt rich features from a variety of images, the VGG16 model is initialized with pre-trained weights from the dataset. The foundation VGG16 model is ready for additional customisation by marking the input shape as (HEIGHT, WIDTH, 3) to match the proportions of the input photos and removing the fully linked layers at the top of the model.

The next step is to freeze the layers of the VGG16 model after setting up the basic model. The pre-trained weights are retained when the layers are set to non-trainable; only the extra layers that are added for classification and fine-tuning will be updated during training. This method assists in utilizing the pre-trained model's information while tailoring it to the particular categorization task at hand.

The goal of the custom model construction phase is to add more classification layers to the VGG16 architecture. To convert the output of the convolutional base into a flat feature vector, flatten layers are typically added. Dense layers are then created in order to carry out the actual classification using the features that were retrieved. The unique needs of the classification issue are taken into account while determining the number of neurons in these Dense layers and the activation functions that are employed.

To get the model ready for training, its architecture must be defined before it can be compiled. This entails defining the evaluation metrics, loss function, and optimizer. For multi-class classification tasks, the SGD optimizer is selected in conjunction with categorical cross-entropy as the loss function. In order to evaluate the model's performance using the training set of data, it is set up to track accuracy

during training.

Lastly, the constructed setup is used to train the model using the supplied image datasets. In order to reduce loss and increase accuracy, the model iterates over the training set of data for a predetermined number of epochs during training. It does this by modifying its weights in accordance with the optimizer and loss function. By adhering to this thorough procedure, the VGG16 model is successfully constructed and prepared for image classification using the tailored classification layers and learned features.

4.4.6 VGG Model Implementation (Without Differential Privacy)

A comprehensive strategy is used in the VGG16 model implementation with differential privacy integration to guarantee the privacy of individual data points during the training phase. Using pre-trained weights from the ImageNet dataset, the VGG16 base model is first set up. The fully connected layers at the top are then excluded, and the input shape is specified to match the dimensions of the input images. By doing this initialization step, you can be sure that the model has a strong base of learned characteristics from a wide range of images. Particular procedures are developed to safeguard the privacy of individual data points in order to include differential privacy in the training process. In order to improve data privacy and stop the model from remembering particular data points, noise is added to the gradients computed during backpropagation. In order to balance privacy protection with model performance, parameters like *noise_multiplier* and *l2_norm_clip* are carefully adjusted to control the amount of noise introduced to the gradients. In addition, the pre-trained weights are retained and not modified during training by freezing the base model layers. By keeping these layers frozen, the differential privacy mechanisms mainly affect the further layers that are added for classification and fine-tuning, making sure that privacy-enhancing methods are used where they are most required. This tactic protects the confidentiality of individual data points while making use of the knowledge stored in the pre-trained model. Custom model creation is like the non-differential privacy scenario, except it extends the VGG16 architecture with more layers for classification. The number of neurons and activation functions in these extra layers, like the Flatten and Dense layers, are designed for precise predictions in accordance with the demands of the particular categorization task. Subsequently, the model is assembled utilizing an optimizer ideal for training differential privacy, like DP-SGD, and set up with proper evaluation metrics and loss functions to track model success. The model incorporates differential privacy techniques while iterating over the given image datasets for training. By including noise into the gradients, the model's updates are prevented from disclosing private information about specific data points, improving privacy protection without sacrificing model performance. Data privacy is well protected by using an all-inclusive approach and including differential privacy strategies into the VGG16 model training process. a makes the model appropriate for applications where privacy protection is a crucial factor.

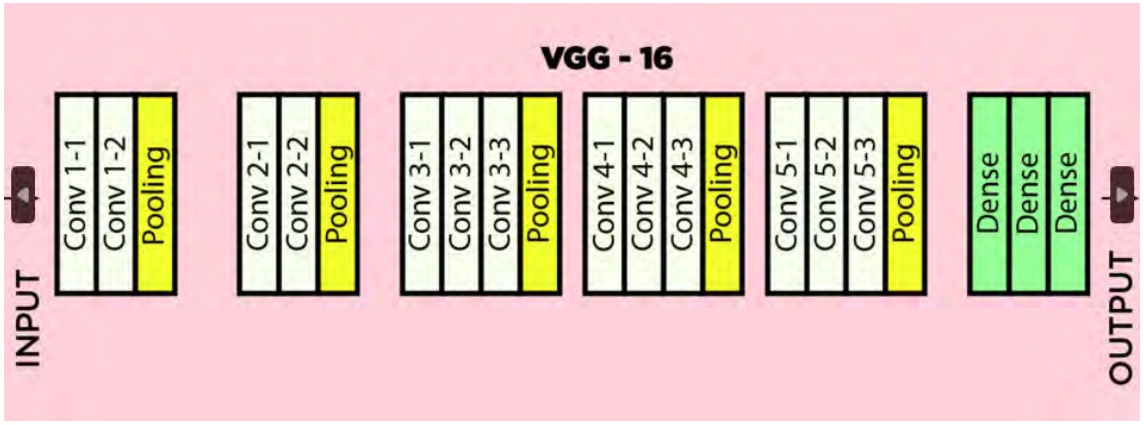


Figure 4.4: VGG16 model architecture

Chapter 5

Result Analysis

5.1 Performance Evaluation Metrics

Several performance evaluation metrics are used to evaluate the efficacy of the AI model created for pneumonia detection using the Kaggle chest X-ray dataset. These metrics guarantee a solid assessment of the model's performance by giving a thorough grasp of its recall, accuracy, precision, F1-score, and confusion matrix.

Accuracy: The percentage of correctly classified samples relative to all samples is known as accuracy. Formula :

$$Acc = \frac{S_c}{T_c}$$

A high accuracy level means that a sizable portion of pneumonia and normal cases are accurately identified by the model.

Precision: By definition, precision is the ratio of the model's total number of true positive predictions to its total number of positive predictions. Formula:

$$P = \frac{T_p}{T_p + F_p}$$

A high degree of precision means that the model has a low false positive rate, which means that normal cases are not frequently misclassified as pneumonia.

Recall: Recall is the percentage of real positive cases that the model correctly detects. Formula:

$$R = \frac{T_p}{T_p + F_n}$$

A high recall rate means that the majority of pneumonia cases are correctly identified by the model, reducing the amount of cases that are missed.

F1-score: The F1-score is defined as the harmonic mean of recall and precision, offering a metric that strikes a balance between the two issues. Formula:

$$[F_1 = 2 * \frac{P * R}{P + R}]$$

A high F1-score shows that the model successfully distinguishes pneumonia cases without over classifying normal cases by maintaining a good balance between precision and recall.

Confusion Matrix: The confusion matrix gives a thorough analysis of the model's performance in classifying each class. It is represented as a N*N square matrix, where N is the number of classes. The confusion matrix provides insights into particular areas where the model might require improvement by helping to visualize the model's performance in terms of true and false positives as well as negatives.

Actual	**Predicted Normal**	**Predicted Pneumonia**
Normal	True Negatives (TN)	False Positives (FP)
Pneumonia	False Negatives (FN)	True Positives (TP)

All of these metrics combined offer a thorough assessment of how well the AI model performs in identifying pneumonia from chest X-ray images. We can make sure that the model performs well in identifying true positive cases, minimizing false positive and false negative rates, and achieving high overall accuracy by employing these metrics. In order to implement a trustworthy and efficient diagnostic tool in medical settings, a comprehensive review is essential.

5.2 Experimental Result Analysis

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.74	0.95	0.83	390
Normal (Class 1)	0.85	0.44	0.58	234
accuracy			0.76	624
macro avg	0.79	0.70	0.71	624
weighted avg	0.78	0.76	0.74	624

Figure 5.1: CNN with Differential Privacy

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.84	0.99	0.91	390
Normal (Class 1)	0.98	0.69	0.81	234
accuracy			0.88	624
macro avg	0.91	0.84	0.86	624
weighted avg	0.89	0.88	0.87	624

Figure 5.2: CNN without Differential Privacy

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.63	0.22	0.32	390
Normal (Class 1)	0.38	0.79	0.51	234
accuracy			0.43	624
macro avg	0.50	0.50	0.41	624
weighted avg	0.53	0.43	0.39	624

Figure 5.3: ResNet-50 with Differential Privacy

	precision	recall	f1-score	support
Pneumonia (Class 0)	0.96	0.35	0.51	390
Normal (Class 1)	0.47	0.97	0.64	234
accuracy			0.58	624
macro avg	0.71	0.66	0.57	624
weighted avg	0.78	0.58	0.56	624

Figure 5.4: ResNet-50 without Differential Privacy

	precision	recall	f1-score	support
NORMAL	0.94	0.40	0.56	234
PNEUMONIA	0.73	0.98	0.84	390
accuracy			0.77	624
macro avg	0.84	0.69	0.70	624
weighted avg	0.81	0.77	0.74	624

Figure 5.5: VGG16 with Differential Privacy

	precision	recall	f1-score	support
NORMAL	0.98	0.42	0.59	234
PNEUMONIA	0.74	0.99	0.85	390
accuracy			0.78	624
macro avg	0.86	0.71	0.72	624
weighted avg	0.83	0.78	0.75	624

Figure 5.6: VGG16 without Differential Privacy

Model	Precision	Recall	F1-Score
CNN (Pneumonia)	0.89	0.97	0.93
CNN (Normal)	0.94	0.79	0.86
CNN with DP (Pneumonia)	0.74	0.95	0.83
CNN with DP (Normal)	0.85	0.44	0.58
Resnet50 (Pneumonia)	0.96	0.35	0.51
Resnet50 (Normal)	0.47	0.97	0.64
Resnet with DP (Pneumonia)	0.63	0.22	0.32
Resnet with DP (Normal)	0.38	0.79	0.51
VGG (Pneumonia)	0.98	0.42	0.59
VGG (Normal)	0.74	0.99	0.85
VGG with DP (Pneumonia)	0.94	0.40	0.56
VGG with DP (Normal)	0.73	0.98	0.84

Table 5.1: Comparison of Different Models with Precision, Recall, and F1-Score

The CNN model with Differential Privacy (DP) for Pneumonia performed better than the CNN model with DP for Normal, according to the comparison in the table. This is the reason why: The CNN for DP (pneumonia) has a recall of 0.95, a figure that is considerably greater than the Normal model's 0.40. Recall quantifies the number of real-world pneumonia cases that the model can accurately identify. A higher recall indicates that pneumonia cases are more accurately detected by the DP-equipped model. The CNN's 0.74 precision for DP (pneumonia) is similar to the 0.94 precision of the Normal model. The precision measure tells us how many positive test results—in this case, pneumonia—are actually positive. The pneumonia model is a better option with DP due to its high recall, even though the Normal model has a higher precision. However, the difference is not statistically significant. When making medical diagnoses, in particular, it is crucial to take the precision vs recall trade-off into account. To prevent false positives, or misdiagnosing normal cases as pneumonia, a high precision is ideal. Missing a case (low recall) in the context of pneumonia, however, might be more dangerous. Consequently, accurately identifying pneumonia cases (high recall) is given priority in the CNN model with DP for Pneumonia.

In conclusion, in the case of differential privacy models, the CNN model with differential privacy was the most successful in detecting pneumonia in this particular situation because it possessed the optimal combination of precision, recall, and F1-score. In general, the models with differential privacy demonstrated a trade-off between recall and precision, with a propensity to misclassify more non-pneumonia images as pneumonia or to miss more real cases.

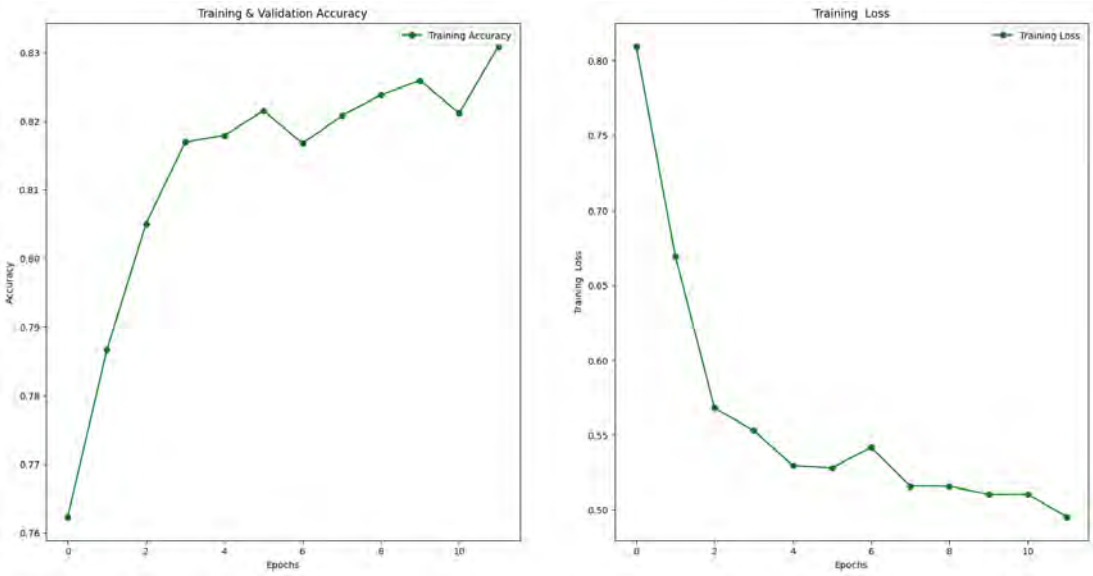


Figure 5.7: CNN with Differential Privacy

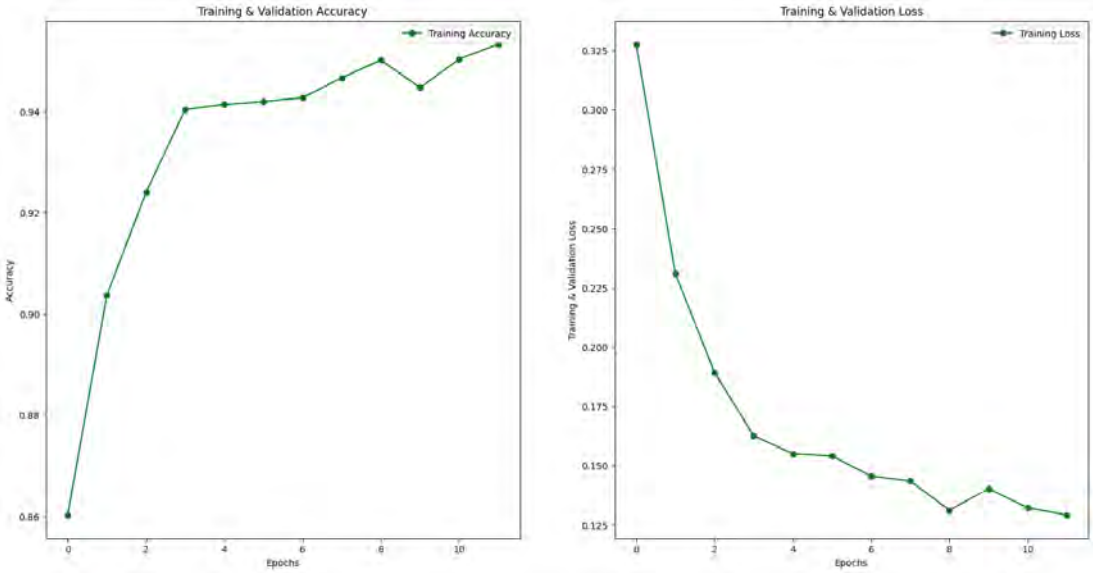


Figure 5.8: CNN without Differential Privacy

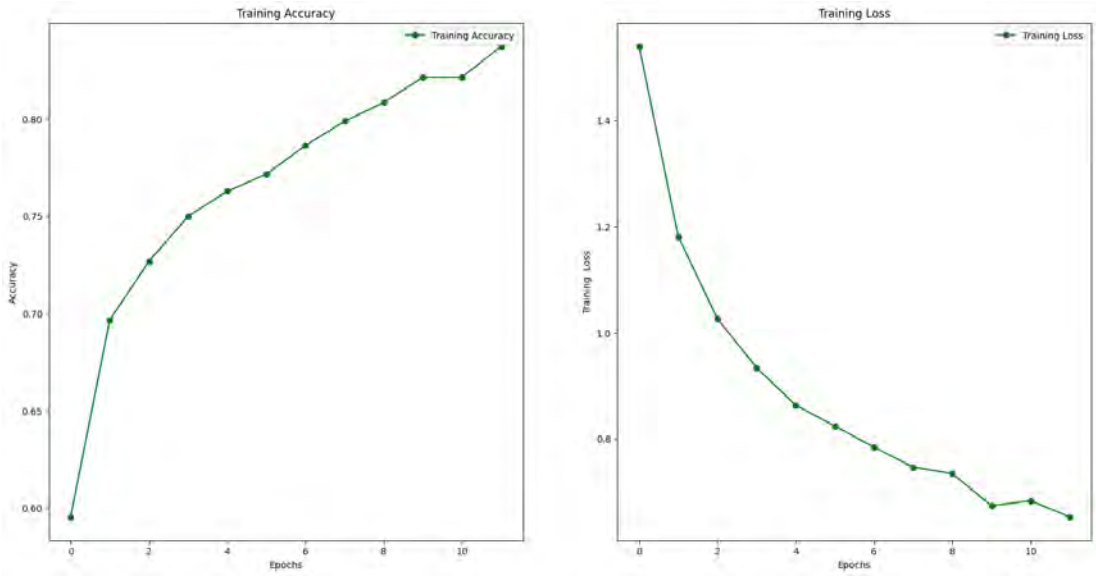


Figure 5.9: ResNet-50 with Differential Privacy

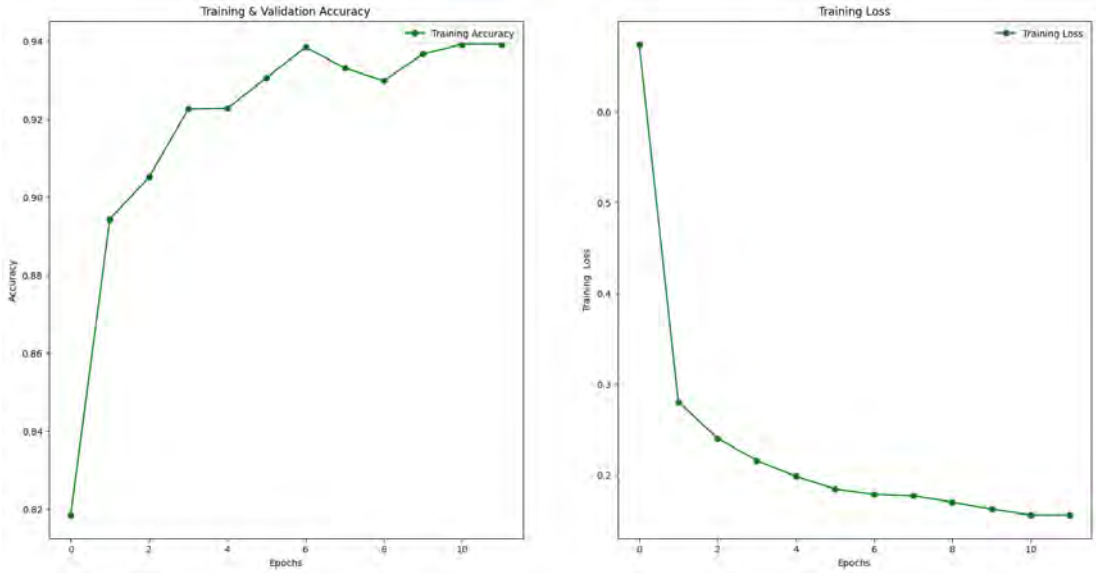


Figure 5.10: ResNet-50 without Differential Privacy

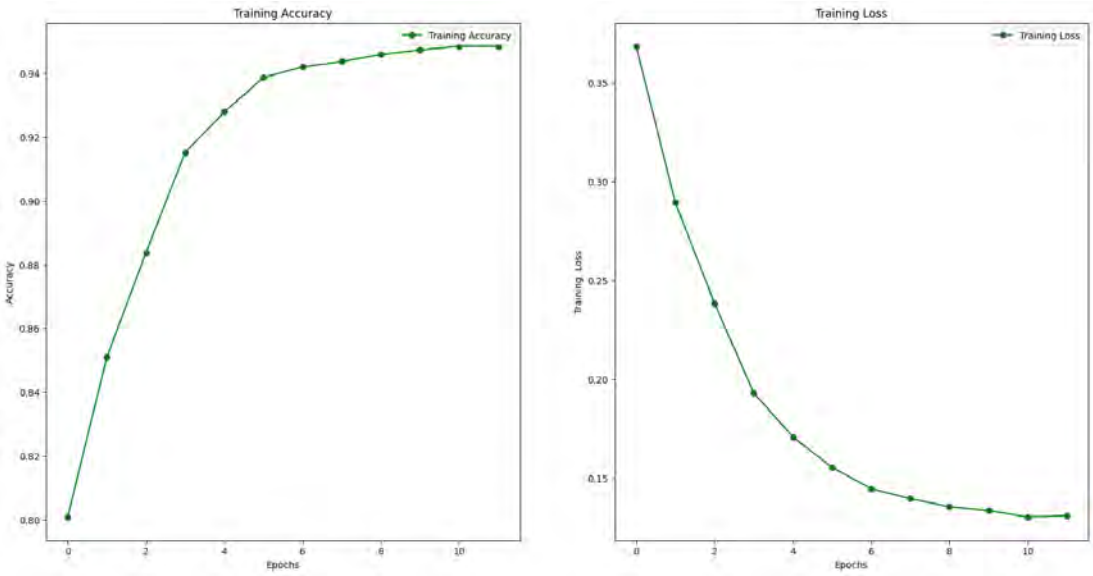


Figure 5.11: VGG16 with Differential Privacy

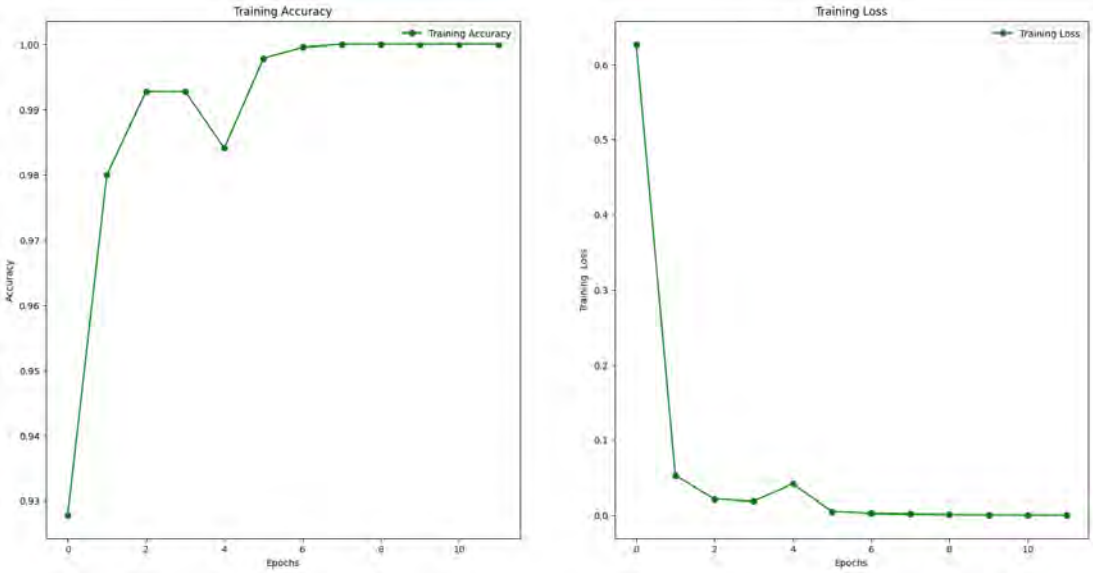


Figure 5.12: VGG16 without Differential Privacy

From the graphs we can see that The model CNN with Differential Privacy worked the best. Where the accuracy was on the increase and the loss was on the decrease.

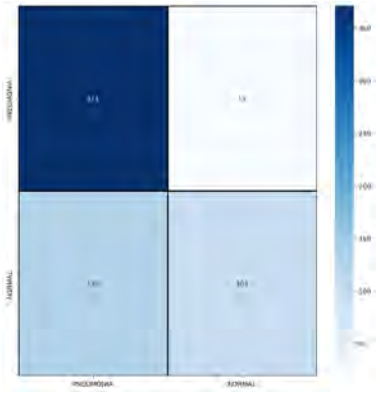


Figure 5.13: CNN with Differential Privacy

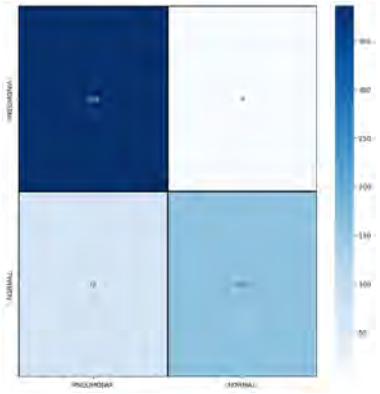


Figure 5.14: CNN without Differential Privacy

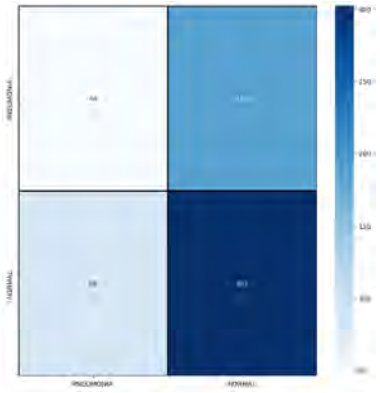


Figure 5.15: ResNet-50 with Differential Privacy

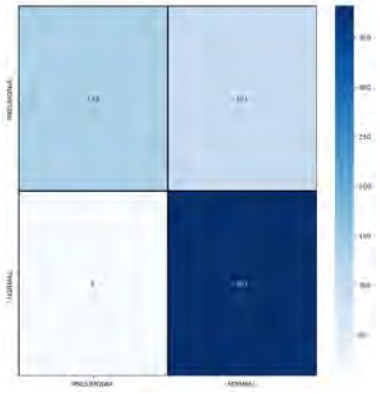


Figure 5.16: ResNet-50 without Differential Privacy

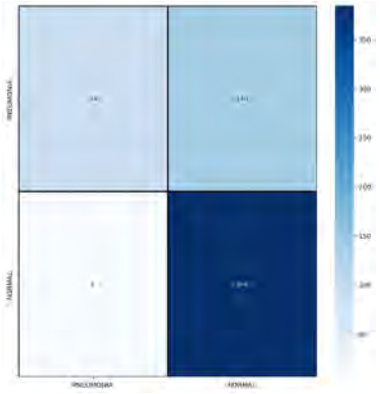


Figure 5.17: VGG16 with Differential Privacy

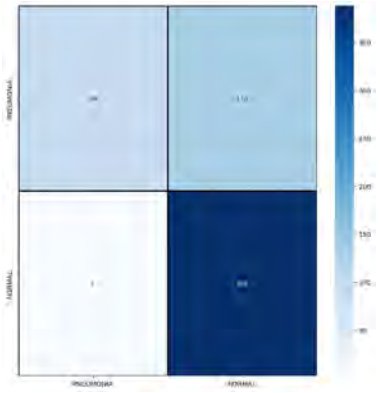


Figure 5.18: VGG16 without Differential Privacy

From the confusion matrix, we can see that The model CNN with Differential Privacy worked the best. Where the accuracy was on the increase and the loss was on the decrease. The accuracy of the CNN model without dp was 90% whereas with dp was 87%. Again Resnet-50 model without dp gives model accuracy of 83% and with dp gives 72% and the VGG model without dp had an accuracy of 70% and with dp had an accuracy of 64%. In this case, it is very sure that the CNN model built by us performed well.

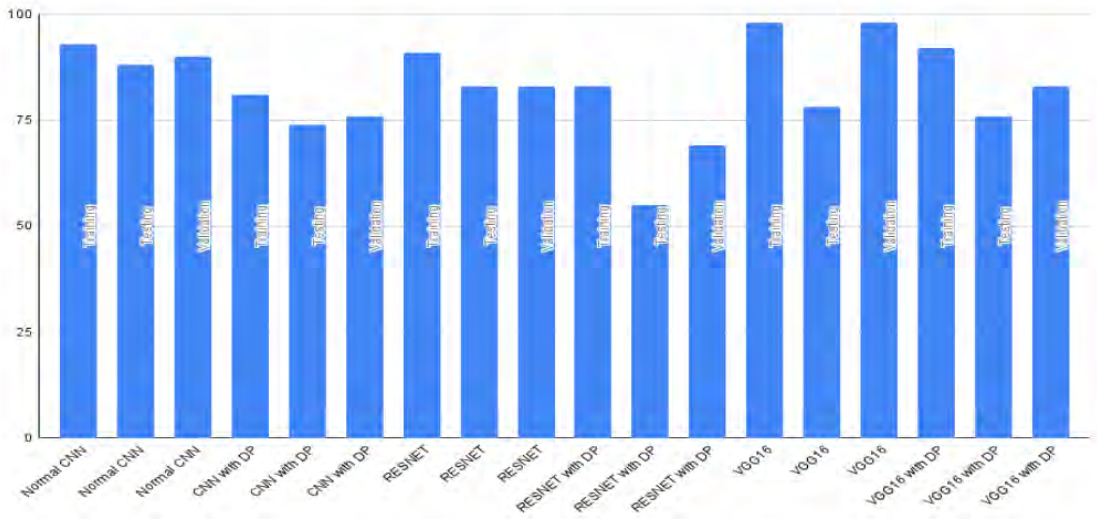


Figure 5.19: CNN with Differential Privacy

We can see that differential privacy significantly drops the accuracy of the models when compared to their models that are not differentially protected. We can also see that the Resnet50 model does not work well with differential privacy. Differentially private Resnet50 model has shown the worst performance with the testing dataset. We got the best testing accuracy with differentially private VGG16. Overall, our proposed model worked best with both differentially private and non-private model.

Chapter 6

Challenges, Limitations, and Future Work

6.1 Challanges

Differential privacy implementation involves a number of complex issues. First of all, maintaining dependencies is important but difficult since it might be difficult to make sure that different libraries like TensorFlow, Keras, and TensorFlow Privacy are installed correctly and compatible. Another problem is data preprocessing, which becomes more difficult when dealing with image data. In order to effectively train models, data quality must be ensured and missing values must be handled carefully. It's crucial but difficult to design an ideal model architecture for the given problem; this involves figuring out each layer's function, adjusting hyperparameters, and balancing model complexity. It can take a lot of computing power to train and optimize the model effectively, which calls for cautious optimization, close attention to metrics like accuracy and loss, and the avoidance of overfitting. Accurately assessing the model's performance, deciphering indicators like precision and recall, and testing the model on hypothetical data are important but difficult processes. Differential privacy and other privacy strategies add complexity and necessitate a thorough comprehension of compliance and privacy issues. To ensure the successful implementation of the machine learning models used, it is imperative to address crucial difficulties such as debugging code, efficiently handling errors, and managing resources like memory and GPU consumption during training and inference.

6.2 Limitation

Differential privacy has a lot of potential for protecting patient data confidentiality, but it also has some drawbacks. It can be difficult to strike a balance between privacy and performance when machine learning models are trying to achieve high levels of privacy at the expense of decreased accuracy. Furthermore, putting differential privacy into practice can result in a large computational overhead, such

as longer processing times and higher memory usage, which is not always possible in healthcare systems, particularly those with constrained computational capacity. The intricacy of accurately putting differential privacy algorithms into practice adds to the complexity since it can be resource-intensive to carefully design and thoroughly test in order to ensure privacy guarantees while maintaining model performance. Scalability is another problem because the extra noise needed to preserve privacy when using differential privacy techniques on big datasets or intricate models can reduce the usefulness of the data. Finally, even though differential privacy offers technical protections, ethical and legal issues still need to be resolved in order to guarantee patient confidentiality and compliance with data protection laws. Differential privacy in healthcare will require ongoing development and adoption, which will require an understanding of and attention to these limitations.

6.3 Future Works

We intend to continue investigating hyperparameter optimization strategies in the near future. These strategies, which include adjusting batch sizes and learning rates, could greatly increase the accuracy and generalization capacity of the model. Improving preprocessing procedures for data, such as augmentation and normalization, would improve the quality of the data and the resilience of the model. A complete examination of the model's performance would also be possible by carrying out exhaustive model evaluation methods, which would include validation metrics and overfitting detection strategies. A larger dataset may be handled by the model more skillfully if scalability and efficiency factors, such as distributed training and model parallelism, were taken into account. Finally, the development of a more privacy-aware machine learning solution would require a deeper exploration of privacy techniques such as differential privacy and an understanding of their impact on model performance and data protection compliance.

Chapter 7

Conclusion

In conclusion, by reading all the related papers on data privacy, especially medical data privacy, we can understand the importance of protecting the privacy of patients in the field. We deeply understand how sensitive personal data is and why most institutes do not want to share their data with research agencies. We also contemplate that differ ential privacy could be a milestone in easing the process of allowing patients' data, as it provides enough security for anonymity and has good performance in regard to ma chine learning tasks. So we plan to use the previously done research as a foundation to further explore the implications of differential privacy in healthcare. We have al ready seen in previous papers that differential privacy together with federated learning helped a lot of researchers develop covid 19 detection using X-rays. So overall, differ ential privacy has a good chance of preserving individual privacy. Reading the articles, we've also seen that there is a trade-off between privacy and performance. We want to find the sweet spot between these two where privacy is maximized and performance does not deteriorate.

Bibliography

[1] L. Sweeney, "Weaving technology and policy together to maintain confidentiality," *The Journal of Law, Medicine & Ethics*, vol. 25, no. 2-3, pp. 98–110, 1997.

[2] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 439–450.

[3] W. Du and Z. Zhan, "Using randomized response techniques for privacy-preserving data mining in: Proceedings of the ninth acm sigkdd international conference on knowledge discovery and data mining, 505–510," *Washington, DC*, 2003.

[4] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and l-diversity," in *2007 IEEE 23rd international conference on data engineering*, IEEE, 2006, pp. 106–115.

[5] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 3–es, 2007.

[6] C. C. Aggarwal and S. Y. Philip, *Privacy-preserving data mining: models and algorithms*. Springer Science & Business Media, 2008.

[7] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*, Springer, 2008, pp. 1–19.

[8] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, IEEE, 2008, pp. 111–125.

[9] K. Chaudhuri and D. Hsu, "Sample complexity bounds for differentially private learning," in *Proceedings of the 24th Annual Conference on Learning Theory*, JMLR Workshop and Conference Proceedings, 2011, pp. 155–186.

[10] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization.," *Journal of Machine Learning Research*, vol. 12, no. 3, 2011.

[11] S. Wang, M. Nassar, M. Atallah, and Q. Malluhi, "Secure and private outsourcing of shape-based feature extraction," in *Information and Communications Security: 15th International Conference, ICICS 2013, Beijing, China, November 20-22, 2013. Proceedings 15*, Springer, 2013, pp. 90–99.

[12] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: A survey and review," *arXiv preprint arXiv:1412.7584*, 2014.

[13] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310–1321.

[14] K. Xu, H. Yue, L. Guo, Y. Guo, and Y. Fang, "Privacy-preserving machine learning algorithms for big data systems," in *2015 IEEE 35th international conference on distributed computing systems*, IEEE, 2015, pp. 318–327.

[15] M. Abadi, A. Chu, I. Goodfellow, *et al.*, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.

[16] D. Apple, "Learning with privacy at scale," *Apple Machine Learning Journal*, vol. 1, no. 8, 2017.

[17] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," *arXiv preprint arXiv:1710.06963*, 2017.

[18] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual review of biomedical engineering*, vol. 19, pp. 221–248, 2017.

[19] O. Choudhury, A. Gkoulalas-Divanis, T. Salonidis, *et al.*, "Differential privacy-enabled federated learning for sensitive health data," *arXiv preprint arXiv:1910.02578*, 2019.

[20] P. Afshar, S. Heidarian, F. Naderkhani, A. Oikonomou, K. N. Plataniotis, and A. Mohammadi, "Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images," *Pattern Recognition Letters*, vol. 138, pp. 638–643, 2020.

[21] M. J. Horry, S. Chakraborty, M. Paul, *et al.*, "Covid-19 detection through transfer learning using multimodal imaging data," *Ieee Access*, vol. 8, pp. 149 808–149 824, 2020.

[22] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, vol. 2, no. 6, pp. 305–311, 2020.

[23] M. A. U. Alam, "Person re-identification attack on wearable sensing," *arXiv preprint arXiv:2106.11900*, 2021.

[24] E. Bozkir, O. Günlü, W. Fuhl, R. F. Schaefer, and E. Kasneci, "Differential privacy for eye tracking with temporal correlations," *Plos one*, vol. 16, no. 8, e0255979, 2021.

[25] H. Mukherjee, S. Ghosh, A. Dhar, S. M. Obaidullah, K. Santosh, and K. Roy, "Deep neural network to detect covid-19: One architecture for both ct scans and chest x-rays," *Applied Intelligence*, vol. 51, pp. 2777–2789, 2021.

[26] A. Ziller, D. Usynin, R. Braren, M. Makowski, D. Rueckert, and G. Kaissis, "Medical imaging deep learning with differential privacy," *Scientific Reports*, vol. 11, no. 1, p. 13 524, 2021.

[27] M. A. Abdou, "Literature review: Efficient deep neural networks techniques for medical image analysis," *Neural Computing and Applications*, vol. 34, no. 8, pp. 5791–5812, 2022.

[28] M. Adnan, S. Kalra, J. C. Cresswell, G. W. Taylor, and H. R. Tizhoosh, "Federated learning and differential privacy for medical image analysis," *Scientific reports*, vol. 12, no. 1, p. 1953, 2022.

[29] T. Kossen, M. A. Hirzel, V. I. Madai, *et al.*, "Toward sharing brain images: Differentially private tof-mra images with segmentation labels using generative adversarial networks," *Frontiers in artificial intelligence*, vol. 5, p. 85, 2022.

[30] Z. Li, X. Xu, X. Cao, *et al.*, "Integrated cnn and federated learning for covid-19 detection on chest x-ray images," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2022.

[31] J. Luo and S. Wu, "Fedsld: Federated learning with shared label distribution for medical image classification," in *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, IEEE, 2022, pp. 1–5.

[32] N. Khalid, A. Qayyum, M. Bilal, A. Al-Fuqaha, and J. Qadir, "Privacy-preserving artificial intelligence in healthcare: Techniques and applications," *Computers in Biology and Medicine*, p. 106 848, 2023.