Differential Privacy Enabled Deep Skin Image Classification Model Development

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

Master of Technology

in

Cryptology and Security

submitted by

Prasun Kumar Mandal

(Roll No. CrS2314)

Under the esteemed guidance of

Dr. Anabik Pal

Deparment of Computer Science

IISER, Berhampur, Odisha, India

Dr. Debrup Chakraborty

Cryptology and Security Research Unit

Indian Statistical Institute, Kolkata

Indian Statistical Institute

Kolkata – 700108, India

July, 2025

Contents

Declaration 3

Certificate 4

Acknowledgements 5

Abstract 6

1 Preliminaries **7**

1.1 Motivation of Data Security 7

1.2 Privacy Issues 8

1.3 Security Enable Deep Learning 13

2 DP Enabled DCNN Formulate **14**

2.1 Basic of Differential Privacy 14

2.2 Deep Networks 16

2.2.1 ResNet-18 Architecture Overview 16

2.2.2 ResNet-50 Architecture Overview 18

3 DP Enabled Deep Model for Skin Disease Classification **20**

3.1 Skin Image Classification : 20

3.2 Literature Review: 20

3.3 Methodology: 21

3.4 Datasets: 29

4 Experiments and Results **30**

4.1 SGD and DP-SGD with ResNet50 model 30

4.2 SGD and DP-SGD with ResNet18 model 31

4.3 ADAM and DP-ADAM with ResNet50 model 33

4.4 ADAM and DP-ADAM with ResNet18 model 35

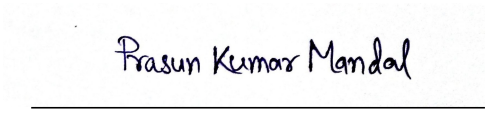
4.5 Observation 36

5 Conclusion and Future Works 37

Declaration

I, **Prasun Kumar Mandal** (Roll No: CrS2314), hereby declare that this report entitled *"Differential Privacy Enabled Deep Skin Image Classification Model Development"*, submitted to Indian Statistical Institute, Kolkata towards the fulfilment of the requirements for the degree of Master of Technology in Cryptology & Security, is an original work carried out by me under the supervision of **Dr. Anabik Pal and Dr. Debrup Chakraborty,** and has not formed the basis for the award of any degree or diploma in this or any other institution. I have sincerely tried to uphold academic ethics and honesty. Whenever a piece of external information or statement or result is used then, that has been duly acknowledged and cited.

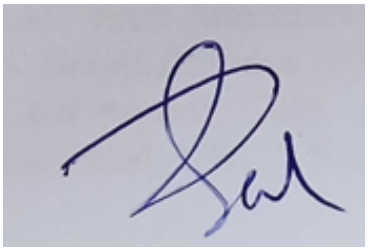
Date: 11 July, 2025



Prasun Kumar Mandal

Certificate

This is to certify that the report entitled ***"Differential Privacy Enabled Deep Skin Image Classification Model Development"***, submitted by **Prasun Kumar Mandal** to the Indian Statistical Institute, Kolkata, for the award of the degree of Master of Technology in Cryptology and Security, is a record of the original, bonafide research work carried out by him under my supervision and guidance.



Dr. Anabik Pal

Assistant Professor, Department of Computer Science

Indian Institute of Science Education and Research, Berhampur

Acknowledgements

The completion of this master's thesis would not have been possible without the invaluable support and guidance of numerous individuals and institutions, to whom I owe my deepest gratitude.

I would like to express my sincere gratitude to my supervisors, Dr. Anabik Pal (IISER Berhampur) and Dr. Debrup Chakraborty (ISI, Kolkata), for their invaluable guidance, support, and encouragement throughout this project. Their expertise and insights have been instrumental in shaping the direction of this study.

I am also thankful to the Indian Statistical Institute, Kolkata, for providing access to a highly stimulating academic environment.

I am also grateful to Indian Institute of Science Education and Research, Berhampur for providing the necessary resources and facilities to conduct this research.

I am equally indebted to my friends, whose thoughtful discussions, moral support, and motivation helped me overcome challenges throughout this endeavor. Their companionship has been a constant source of inspiration.

Finally, I would like to thank everyone who, directly or indirectly, contributed to the successful completion of this thesis. Any errors or shortcomings that remain are solely my responsibility.

Date: 11 July, 2025

Prasun Kumar Mandal

Prasun Kumar Mandal

Abstract

In the era of big data, the explosive growth in data volume has significantly accelerated the development of deep learning. Deep learning is the most promising area of AI, yielding significant advancements in medical image classification. However, healthcare data contains important sensitive information and so privacy and security are crucial to preventing unauthorized access. Note that there are several data protection rules from multiple regulations to penalize any kind of data security violation, for example, the data protection principles (Article 5.1-2) and the data protection by design and by default (Article 25) of the General Data Protection Regulation from the European Union. It is mandatory to follow such data regulations in developing and deploying deep models. Traditional deep learning models are vulnerable to several types of attacks, including membership inference attacks, where an adversary determines whether a specific data point was used in training; model extraction attacks, where attackers attempt to replicate the functionality of a trained model and reconstruction attacks, which aim to recover original training data from model outputs.

To mitigate data privacy leakage in deep learning models, this dissertation will focus on development of "Differential Privacy enabled deep model" that can deal with the privacy leakage from the trained model. In this research primarily gradient clipping-based deep optimization algorithms (such as DP-SGD, DP-Adam) will experiment with.

Automated classification of dermatological images will be the chosen application field for this research. Literature shows that several deep models exist which deal with dermatological image analysis and produce promising performance. However, the performance drop has not yet been explored adequately when such a model was trained with an optimization algorithm that preserves differential privacy. A number of deep neural networks is experimented with to assess performance degradation with the chosen secure training mechanism. Finally, this dissertation aims to develop a novel technique to build differential privacy enabled skin model. This dissertation is utilizing publicly available dermatological image datasets like ISIC 2018.

Chapter 1

Preliminaries

In the modern day, the rapid increase in data volume has significantly accelerated the development of deep learning. As a leading subfield of machine learning, deep learning has achieved state-of-the-art performance across various domains such as images, sounds, and texts. Its applications include social network analysis [13] , bioinformatics [4], medicine and healthcare [5] and more.

Despite its success, the use of sensitive healthcare data introduces serious concerns regarding privacy and security. Numerous regulations, such as the General Data Protection Regulation (GDPR) [1] of the European Union, impose strict requirements for data handling. For instance, GDPR's Article 5.1-2 outlines key principles for data protection, and Article 25 mandates data protection by design and by default.

Traditional deep models, however, are susceptible to various attacks. These include membership inference attacks [14], which reveal whether a particular record was part of the training data; model extraction attacks [15], where adversaries try to copy the model's behavior; and reconstruction attacks [11], which attempt to regenerate original training data based on the model's outputs.

Alignment with these above laws is essential when developing and deploying deep learning models.

1.1. Motivation of Data Security

The motivation behind data security lies in the critical need to protect sensitive and valuable information from unauthorized access, misuse or alteration. In today's digital

world, individuals, organizations and governments increasingly rely on data for operations, decision-making and communication. Ensuring data confidentiality, integrity and availability is essential to prevent identity theft, financial fraud data breaches and loss of trust. Data security also plays a vital role in maintaining regulatory compliance, safeguarding intellectual property and preserving organizational reputation.

In the healthcare sector, data security becomes even more significant due to the highly sensitive nature of patient records, diagnoses and treatment histories. Protecting healthcare data is essential to maintain patient privacy, prevent medical identity theft and ensure accurate and uninterrupted medical care. Security violation in healthcare systems can lead to legal liabilities and erosion of public trust. Therefore, securing healthcare data is not only a matter of compliance with standards like Health Insurance Portability and Accountability Act(HIPAA) [3] but also a fundamental requirement for safe, ethical and efficient delivery of medical services.

1.2. Privacy Issues

In the context of deep learning, a privacy issue arises when sensitive information used in training or inference is at risk of being exposed or misused. An adversary refers to an entity that actively attempts to exploit the model or its data. The goal of such an adversary is typically to manipulate the model's behavior, extract confidential information or deceive the system by leveraging its vulnerabilities. These attacks can lead to significant violates of privacy, especially when models are trained on personal, financial or medical data.

Particularly in classification tasks, several privacy attacks present significant threats to the confidentiality of training data and the integrity of the model. These attacks are typically carried out by adversaries seeking to exploit the model's behavior or internal parameters to gain unauthorized access to sensitive information. There are three popular attack such as Membership Inference Attack [14], Model Extraction Attack [15] and Reconstruction Attack [11].

Membership Inference Attack

In recent years, Deep Learning (DL) models have demonstrated remarkable performance across various domains such as healthcare, finance, image recognition, and natural language processing. However, as these models are increasingly trained on sensitive data, privacy concerns have emerged. One of the most widely studied privacy threats in this context is the Membership Inference Attack (MIA). A Membership Inference Attack is a type of privacy attack in which an adversary attempts to determine whether a specific data point was used in the training dataset of a deep learning model. In simpler terms, the attacker tries to infer the "membership status" of a given instance—whether it was part of the model's training set or not—by observing the model's predictions or behavior.

This attack becomes particularly concerning when models are trained on sensitive datasets, such as patient medical records, user profiles, or financial transactions. Even if the actual data is not shared publicly, the trained model may still leak information about its training set through overfitting or response patterns.

The attacker typically queries the target model with a specific input and observes its confidence score, probability vector or output class. Based on these outputs, the attacker analyzes differences in the model's behavior on training data versus unseen data.

The attack may be executed using the following steps:

Black-box access: The attacker has query access to the model's predictions but not to its internal parameters.

Shadow model training: The attacker trains one or more models (shadow models) on a dataset similar in distribution to the target model's training data to simulate its behavior.

Attack model training: Using outputs from the shadow models, the attacker trains a classification attack model that learns to distinguish whether an input was in the training set or not.

Model Extraction Attack

In modern deep learning systems, particularly those deployed via public APIs or cloud-based platforms, security threats have evolved to include attacks not just on data, but also on the models themselves. One such threat is the Model Extraction Attack [15],

in which an adversary attempts to extract the entire or partial ML model architecture, weights, or parameters.
Model extraction is particularly dangerous because it allows the attacker to steal the model's functionality without accessing the original training data or internal code. This undermines the intellectual property of the model owner, as models often represent the result of significant investment in data collection, feature engineering, and computational resources. Once the model is extracted, it can be misused.

Model Extraction Attack: Design

The primary goal of a Model Extraction Attack [15] is to replicate or approximate the functionality of a target machine learning model M by using only input-output interactions. The attacker builds a surrogate model M' that mimics the behavior of M without access to the original training data, model weights, or internal architecture.

Assumptions

- The attacker has **black-box access** to the model: they can send input queries to M and receive predictions.
- The internal parameters, training dataset, and architecture of M are not disclosed.
- The attacker may receive either hard labels or soft probability outputs from M.

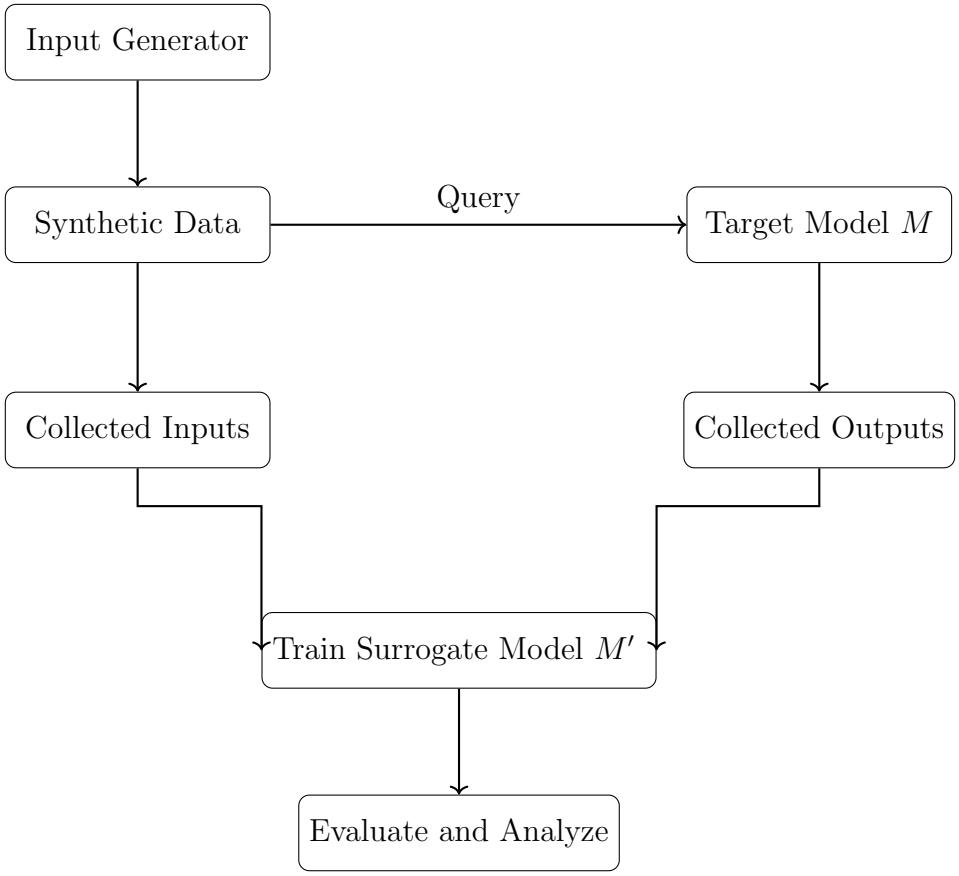
Attack Pipeline

1. **Input Generation:** Generate or sample a set of inputs x_i from the data distribution(public dataset or synthetic data).
2. **Query the Target Model:** For each x_i, query the model to get $y_i = M(x_i)$. These may be predicted class labels or probability vectors.
3. **Construct a Surrogate Dataset:** Build a dataset $D' = \{(x_i, y_i)\}$ from the collected queries and responses.
4. **Train the Surrogate Model:** Use D' to train a surrogate model M' that replicates the decision boundaries of M.

5. **Evaluation:** Measure how well M' approximates M by comparing their predictions on new inputs (fidelity) or using accuracy if ground truth labels are known.

Architecture Overview

Below is a schematic flow of a model extraction attack:



Reconstruction Attack

The adversary's goal in a reconstruction attack [11] is to recreate the original input that produced a specific output prediction by exploiting the internal behavior of the model. By reverse-engineering the relationship between the model's outputs and its corresponding inputs, the adversary can potentially extract sensitive information and gain insights into the original data points that were used during the model's training process.

Assumptions

For a successful reconstruction attack, the following assumptions are generally made:

- The attacker has **black-box access** to the target model, meaning they can query the model with inputs and observe its outputs.
- The attacker may also know the **data distribution** or feature space from which inputs are drawn.
- The model's outputs are sufficiently informative (e.g., probability scores, confidence levels, or internal embeddings).
- In some cases, the attacker may have access to intermediate layers or gradients (white-box setting).

Design Steps of the Attack

The reconstruction attack typically follows these core stages:

1. Querying the Target Model The attacker sends multiple input queries to the target model. These queries can be randomly generated, crafted based on prior knowledge, or sampled from known data distributions. The model returns output predictions for each query.

2. Capturing Model Outputs The attacker collects the model's responses, which may include softmax probability vectors, logits, class predictions, or embeddings. These outputs often encode rich information about the data the model was trained on.

3. Formulating an Inverse Problem The core of the attack is to solve an inverse problem:

$$\hat{x} = \arg\min_{x} \mathcal{L}(f(x), y)$$

where $f(x)$ is the model's output for input x, and y is the observed output (e.g., label or probability vector). The attacker uses optimization techniques to find the input $\hat{x}$ that most likely caused the model to produce output y.

4. Output Reconstructed Data The final reconstructed data $\hat{x}$ may resemble original training samples, capturing identifiable patterns or features. These reconstructed instances can leak sensitive attributes such as patient faces, financial records, or textual content.

1.3. Security Enable Deep Learning

To mitigate privacy threats such as membership inference, model extraction and reconstruction attacks, several robust privacy-preserving techniques have been developed. Among the most effective is Differential Privacy (DP) proposed by Dwork et al [6]in 2006 , which introduces mathematical guarantees to limit the influence of any individual data point on the model. By injecting carefully calibrated noise into gradients or outputs, DP enables model training while preserving individual-level privacy.

In distributed environments, where data is stored across multiple devices or institutions, then we use Federated Learning (FL) [12] for data privacy. It allows models to be trained locally on decentralized data without ever sharing the raw data itself. Only model updates are exchanged, thereby reducing the risk of data leakage.

To further enhance security, techniques such as Homomorphic Encryption (HE) and Secure Multi-Party Computation (SMPC) are also used. Homomorphic Encryption allows computation to be performed directly on encrypted data, enabling inference without decryption. On the other hand, SMPC allows multiple parties to jointly compute a function over their inputs while keeping those inputs private, making it suitable for collaborative learning settings where data sharing is restricted.

Here we used Differential privacy for secure the deep learning model.

Chapter 2

DP Enabled DCNN Formulate

Diferential privacy [7] constitutes a robust standard for privacy guarantees for algorithms on aggregate databases. It is defined using the concept of neighboring databases, which is specific to the application. For example, each training dataset in our research is a collection of image-label pairs; two sets are considered nearby if they differ in a single entry, meaning that one image-label combination appears in one set but not in the other.

2.1. Basic of Differential Privacy

A. Defination

Definition 1. A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is *ε-differential privacy* if for any two adjacent inputs $D_1, D_2 \in \mathcal{D}$ and for any subset of outputs $O \subseteq \mathcal{R}$, it holds that:

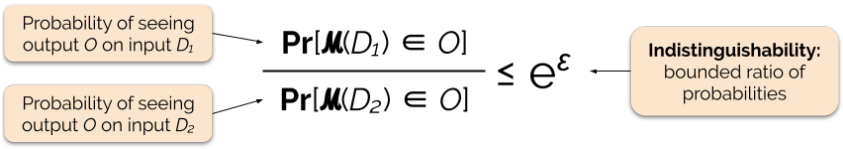


Figure 2.1: Formal Definition of Differential Privacy

So, ε-differential privacy implies that a change to one entry in a database only creates a small change in the probability distribution of the outputs of measurements, as seen by the attacker.

Definition 2. A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-*differential privacy* if for any two adjacent inputs $d, d' \in \mathcal{D}$ and for any

subset of outputs $S \subseteq \mathcal{R}$, it holds that:

$$\Pr[\mathcal{M}(d) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(d') \in S] + \delta$$

The original definition of ε-differential privacy, as introduced by Dwork et al. [7] , does not include the additive term δ. The (ε, δ)-differential privacy is a relaxed version that allows for a small probability δ (preferably much less than $\frac{1}{|d|}$).
If is $\delta = 0$, then the randomized mechanism M gives-differential privacy by its strictest defnition.

B. Sensitivity

Let $f : N^{|X|} \to R^d$ be an arbitrary d-dimensional function, and define its ℓ_2 sensitivity to be $\Delta_2 f = \max_{\text{adjacent } x,y} \|f(x) - f(y)\|_2$.

C. Gaussian Mechanism

For instance, the **Gaussian noise mechanism** is defined as:

$$\mathcal{M}(d) = f(d) + \mathcal{N}(0, \Delta_2 f^2)$$

where $\mathcal{N}(0, \Delta_2 f^2)$ denotes a normal (Gaussian) distribution with mean 0 and standard deviation $\Delta_2 f$.

A single application of the Gaussian mechanism to a function f with **sensitivity** $\Delta_2 f$ satisfies (ϵ, δ)-differential privacy if:

$$\delta \geq \frac{4}{5} \exp\left(-\frac{\epsilon^2}{2}\right)$$

This provides a bound on δ in terms of ϵ, given the standard deviation of the Gaussian noise is properly scaled according to the sensitivity of the function and $\epsilon < 1$ [See, Theorem 1]

D. Privacy Loss

Privacy loss is a random variable that reflects the amount of information leaked by the mechanism due to the added noise. As described by Dwork et al., for neighboring datasets d and d', a differentially private mechanism $\mathcal{M}(D) \in \mathbb{R}$, an auxiliary input *aux*, and a possible outcome $o \in \mathbb{R}$, the privacy loss at outcome o is defined as:

$$c(o; \mathcal{M}, \text{aux}, d, d') = \log \frac{\Pr[\mathcal{M}(\text{aux}, d) = o]}{\Pr[\mathcal{M}(\text{aux}, d') = o]} \tag{2.1}$$

Privacy loss is typically calculated at each step during training or algorithm execution, and its accumulation over multiple steps is used to estimate the overall privacy budget consumed. This cumulative tracking is performed by a **privacy accountant**, which enables the mechanism to maintain a global (ϵ, δ)-DP guarantee.

Theorem 1 *Let $\varepsilon \in (0, 1)$ be arbitrary. For $c^2 > 2\ln(1.25/\delta)$, the Gaussian Mechanism with parameter $\sigma \geq c \cdot \frac{\Delta_2 f}{\varepsilon}$ is (ε, δ)-differentially private.*

This theorem proof by Dwork et al. [7] in Appendices A.

2.2. Deep Networks

2.2.1. ResNet-18 Architecture Overview

ResNet-18 is a widely used convolutional neural network architecture that belongs to the family of residual networks (ResNets) [9]. It was designed to address the degradation problem observed in very deep networks, where increasing the depth of the model leads to worse performance due to vanishing gradients. ResNet overcomes this by introducing *skip connections*, which allow the network to learn residual mappings rather than direct transformations.

Each residual block in ResNet-18 contains two 3×3 convolutional layers, each followed by batch normalization and a ReLU activation. The input to each block is added to the output through a shortcut connection. If the input and output dimensions differ, a 1×1 convolution is used for projection to match dimensions before addition. When dimensions match, the shortcut is simply an identity connection.

The overall architecture of ResNet-18 includes:

1. An initial convolutional layer with a 7×7 kernel and stride 2, followed by batch normalization, ReLU activation, and 3×3 max pooling.

2. Four stages of residual blocks:

   - `Conv2_x`: two blocks with 64 filters.
   - `Conv3_x`: two blocks with 128 filters.
   - `Conv4_x`: two blocks with 256 filters.
   - `Conv5_x`: two blocks with 512 filters.

3. A global average pooling layer.

4. A fully connected (dense) layer for classification.

The residual learning framework enables effective training of deep networks by allowing gradients to propagate more smoothly. This architecture has proven successful in a variety of image classification tasks and is particularly suitable for medical imaging applications due to its balanced depth and computational efficiency.

Layer Name	**Output Size**	**Layer Configuration**	**Parameters**
Input	224×224×3	RGB Image	-
Conv1	112×112×64	7×7 conv, stride=2, padding=3	9,408
BatchNorm1	112×112×64	Batch Normalization	128
ReLU	112×112×64	ReLU Activation	-
MaxPool	56×56×64	3×3 MaxPool, stride=2	-
Conv_2x	56×56×64	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	147,456
Conv_3x	28×28×128	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	525,312
Conv_4x	14×14×256	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	2,097,152
Conv_5x	7×7×512	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	8,388,608
AvgPool	1×1×512	7×7 Global Average Pool	-
FC	1×7	Fully Connected (7 classes)	3,584
Softmax	1×7	Softmax Activation	-

Table 2.1: ResNet-18 Architecture Overview

2.2.2. ResNet-50 Architecture Overview

ResNet50 is a powerful image classification model that can be trained on large datasets and achieve state-of-the-art results. One of its key innovations is the use of residual connections, which allow the network to learn a set of residual functions that map the input to the desired output. These residual connections enable the network to learn much deeper architectures than was previously possible, without suffering from the problem of vanishing gradients.

The architecture of ResNet50 is divided into four main parts: the convolutional layers, the identity block, the convolutional block, and the fully connected layers. The convolutional layers are responsible for extracting features from the input image, while the identity block and convolutional block are responsible for processing and transforming these features. Finally, the fully connected layers are used to make the final classification.

The convolutional layers in ResNet50 consist of several convolutional layers followed by batch normalization and ReLU activation. These layers are responsible for extracting features from the input image, such as edges, textures, and shapes. The convolutional layers are followed by max pooling layers, which reduce the spatial dimensions of the feature maps while preserving the most important features.

The identity block and convolutional block are the key building blocks of ResNet50. The identity block is a simple block that passes the input through a series of convolutional layers and adds the input back to the output. This allows the network to learn residual functions that map the input to the desired output. The convolutional block is similar to the identity block, but with the addition of a 1x1 convolutional layer that is used to reduce the number of filters before the 3x3 convolutional layer.

The final part of ResNet50 is the fully connected layers. These layers are responsible for making the final classification. The output of the final fully connected layer is fed into a softmax activation function to produce the final class probabilities.

Differentially Private Deep Learning Mechanisms

There are several representative deep learning approaches that incorporate differential privacy. Differential privacy provides a formal guarantee that the outputs of a trained model remain statistically similar even when trained on neighboring datasets—datasets

differing by just one individual record. This helps to mitigate privacy leakage risks in both white-box and black-box attack settings.

Most of the mechanisms discussed aim to defend against *membership inference attacks*, as introduced earlier. Depending on the stage of the data processing pipeline in deep learning, the application of differential privacy can be categorized into three primary layers:

- **Input layer**: Noise is added directly to the input data or features.
- **Hidden layer**: Noise is injected during intermediate computations such as gradients or activations.
- **Output layer**: The final model output or parameters are privatized before release.

Chapter 3

DP Enabled Deep Model for Skin Disease Classification

3.1. Skin Image Classification :

Skin diseases are among the most common health conditions globally, affecting millions of individuals across all age groups. Accurate and timely diagnosis is essential for effective treatment and improved patient outcomes, particularly in the case of life-threatening conditions such as melanoma. However, experienced dermatologists is often limited, especially in rural or resource-constrained regions, which can delay diagnosis and treatment.

In this context, skin image classification, powered by advances in deep learning and computer vision, has emerged as a promising technique for the automated identification of various dermatological conditions using dermoscopic or clinical images. By analyzing the visual patterns and features present in skin lesions, such systems can assist clinicians in detecting diseases such as melanoma, basal cell carcinoma, eczema, and psoriasis at an early stage. These automated tools not only help bridge the gap in healthcare accessibility but also reduce the workload on medical professionals, enabling faster and more consistent decision-making in dermatological diagnosis

3.2. Literature Review:

The literature reveals that numerous deep learning models have been developed for dermatological image analysis, many of which demonstrate promising classification performance on benchmark datasets. These models, typically trained using standard optimization

techniques, are capable of achieving accuracy levels comparable to expert dermatologists in certain tasks such as skin lesion classification.

However, the impact on model performance when training is carried out using optimization algorithms that enforce differential privacy. While differential privacy provides formal guarantees for preserving individual data confidentiality, which may degrade model accuracy. Despite its growing importance in privacy-sensitive medical domains, few studies have systematically analyzed the trade-off between privacy preservation and diagnostic performance in the context of deep dermatological models.

3.3. Methodology:

In this research primarily gradient clipping-based deep optimization algorithms (such as DP-SGD, DP-Adam) will experiment with various CNN models like ResNet18, ResNet50.

This dissertation proposes a privacy-preserving framewor under Differential Privacy setting. To ensure privacy in SGD, the gradients are clipped and then noise is added to them.

Algorithm 1 Differentially private SGD (Outline) [2]

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.
Initialize θ_0 randomly
for $t \in [T]$ **do**
    Take a random sample L_t with sampling probability L/N
    Compute gradient
    for each $i \in L_t$ **do**
        $g_t(x_i) \leftarrow \nabla_\theta \mathcal{L}(\theta_t, x_i)$
    end for
    Clip gradient
    $\bar{g}_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{\|g_t(x_i)\|_2}{C})$
    Add noise
    $\tilde{g}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$
    Descent
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{g}_t$
end for
Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

Algorithm 2 Differentially Private Adam (Outline)

Input: Dataset $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N} \sum_i \mathcal{L}(\theta, x_i)$
Hyperparameters: Learning rate η_t, noise scale σ, batch size L, gradient norm bound C, exponential decay rates β_1, β_2, smoothing term ϵ_1
Initialize θ_0, $m_0 = 0$, $v_0 = 0$
for $t = 1$ to T **do**
    Sample a minibatch L_t with probability L/N
    Compute per-example gradients:
    for each $i \in L_t$ **do**
        $g_t(x_i) \leftarrow \nabla_\theta \mathcal{L}(\theta_t, x_i)$
        Clip: $\bar{g}_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{\|g_t(x_i)\|_2}{C})$
    end for
    Add Gaussian noise:
    $\tilde{g}_t \leftarrow \frac{1}{L} \left(\sum_i \bar{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \right)$
    Update biased first and second moment estimates:
    $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1) \tilde{g}_t$
    $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2) \tilde{g}_t^2$
    Bias correction:
    $\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$
    $\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$
    Descent step:
    $\theta_{t+1} \leftarrow \theta_t - \eta_t \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon_1)$
end for
Output: Final model θ_T and total privacy loss (ε, δ) using privacy accounting.

Norm Clipping:

To prove the differential privacy guarantees of Algorithm 1 & 2, it is necessary to bound the influence of any single example on the gradient $\tilde{g}_t$. Since gradients can be arbitrarily large, we apply ℓ_2-norm clipping to each individual gradient. Specifically, a gradient vector g is replaced by $g / \max(1, \frac{\|g\|_2}{C})$, where C is the clipping threshold. This operation leaves g unchanged when $\|g\|_2 \leq C$, and scales it down proportionally when $\|g\|_2 > C$, ensuring that all gradients have a maximum norm of C. We remark that gradient clipping of this form is a popular ingredient of SGD for deep networks for non-privacy reasons, though in that setting it usually suffices to clip after averaging.

Per-layer and time-dependent parameters:

In Algorithm 1 & 2, all parameters are grouped into a single input to the loss function $\mathcal{L}(.)$. However, for multi-layer neural networks, it is beneficial to treat each layer individually.

This enables the use of distinct clipping thresholds C and noise scales σ for each layer. Moreover, both clipping and noise parameters may vary with the number of the training step t.

Lots:

To compute the gradient of the loss function $\mathcal{L}$, we take the average of the gradients over a group of examples. This averaged gradient serves as an unbiased estimator, and its variance decreases rapidly as the group size increases. We refer to such a group as a lot to differentiate it from the computational grouping that is commonly called a batch. To reduce memory usage, the batch size may be set significantly smaller than the lot size L, which is a tunable parameter of the algorithm. Computation is carried out in smaller batches, which are then combined into a lot before noise is added. For computational efficiency, examples are randomly shuffled and then divided into appropriately sized batches and lots. However, for theoretical analysis, we assume that each lot is formed by including each data point independently with probability $q = L/N$, where N is the total number of examples in the dataset.

Privacy Accounting:

For algorithms like DP-SGD and DP-Adam, a critical aspect is determining the total privacy loss incurred during training. Rényi Differential Privacy (RDP) enables the use of a privacy accountant—a mechanism that tracks the privacy cost at each data access and accumulates it over the course of training.

Since each training step involves computing gradients across multiple layers, the accountant aggregates the privacy cost associated with all of them, thereby providing an estimate of the overall privacy loss throughout the training process.

Theorem 2 *Algorithm 1 & 2 preserve Differential Private i.e, the Gaussian mechanism satisfies*(ϵ, δ)*-DP*

For single-step updates:

A mechanism M satisfies (ϵ, δ)-DP if for all measurable sets S, for all neighboring datasets D and D'

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta$$

Now let two Gaussian distributions are

$$P = \mathcal{N}(\mu, \sigma^2) \quad \text{(output distribution for dataset } D\text{)}$$
$$Q = \mathcal{N}(\mu', \sigma^2) \quad \text{(output for neighboring dataset } D'\text{)}$$

Therefore, the difference in means is $d = |\mu - \mu'| \leq \Delta$, where Δ is the Sensitivity. The privacy loss at point x can be written as:

$$L(x) = \log \frac{P(x)}{Q(x)}$$

From the PDF of Gaussian distribution

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$
$$Q(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu')^2}{2\sigma^2}\right)$$

$$\begin{aligned}
L(x) &= \log \frac{P(x)}{Q(x)} \\
&= \log \left(\frac{\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}{\exp\left(-\frac{(x-\mu')^2}{2\sigma^2}\right)} \right) \\
&= -\frac{(x-\mu)^2}{2\sigma^2} + \frac{(x-\mu')^2}{2\sigma^2} \\
&= \frac{(x-\mu')^2 - (x-\mu)^2}{2\sigma^2} \\
&= \frac{(x^2 - 2x\mu' + \mu'^2) - (x^2 - 2x\mu + \mu^2)}{2\sigma^2}
\end{aligned}$$

$$= \frac{-2x(\mu' - \mu) + (\mu'^2 - \mu^2)}{2\sigma^2}$$
$$= \frac{-2xd + (\mu'^2 - \mu^2)}{2\sigma^2}$$

Now, consider x is a random draw from $\mathcal{N}(\mu, \sigma^2)$, we have:

$$x = \mu + Z \quad \text{where } Z \sim \mathcal{N}(0, \sigma^2)$$

Then $L(x)$ will be as follows:

$$\begin{aligned} L(x) &= \frac{-2(\mu + Z)d + \mu'^2 - \mu^2}{2\sigma^2} \\ &= \frac{-2\mu d - 2Zd + \mu'^2 - \mu^2}{2\sigma^2} \\ &= \frac{d^2 - 2Zd}{2\sigma^2} \quad \text{as } \mu'^2 - \mu^2 - 2\mu d = (\mu' - \mu)^2 = d^2 \\ &= \frac{d^2}{2\sigma^2} - \frac{Zd}{\sigma^2} \end{aligned}$$

As $Z \sim \mathcal{N}(0, \sigma^2)$, so $-\frac{Zd}{\sigma^2}$ is Gaussian with mean 0 and variance $\frac{d^2}{\sigma^2}$. Therefore, we have:

$$L(x) \sim \mathcal{N}\left(\mu_L = \frac{d^2}{2\sigma^2}, \quad \sigma_L^2 = \frac{d^2}{\sigma^2}\right)$$

Since $L(x)$ is Gaussian, the equivalent standard normal variable (mean 0, variance 1) will look like

$$\begin{aligned} \Pr[L(x) \geq \varepsilon] &= \Pr\left[\frac{L(x) - \mu_L}{\sigma_L} \geq \frac{\varepsilon - \mu_L}{\sigma_L}\right] \\ &= \Pr\left[\frac{\frac{d^2}{2\sigma^2} - \frac{Zd}{\sigma^2} - \frac{d^2}{2\sigma^2}}{\frac{|d|}{\sigma}} \geq \frac{\varepsilon - \frac{d^2}{2\sigma^2}}{\frac{|d|}{\sigma}}\right] \\ &= \Pr\left[\frac{-\frac{Zd}{\sigma^2}}{\frac{|d|}{\sigma}} \geq \frac{\varepsilon - \frac{d^2}{2\sigma^2}}{\frac{|d|}{\sigma}}\right] \\ &= \Pr\left[-\frac{Z}{\sigma} \geq \frac{\varepsilon - \frac{d^2}{2\sigma^2}}{\frac{|d|}{\sigma}}\right] \\ &= \Pr\left[Z_1 \geq \frac{\varepsilon\sigma}{|d|} - \frac{|d|}{2\sigma}\right], \quad \text{where } Z_1 \sim \mathcal{N}(0, 1) \end{aligned}$$

So the standardized z-score is:

$$\frac{\varepsilon\sigma}{|d|} - \frac{|d|}{2\sigma}$$

According to the Gaussian tail bound:

$$\Pr[Z_1 \geq z] \leq \exp\left(-\frac{z^2}{2}\right)$$

To satisfy (ε, δ)-DP, we need:

$$\exp\left(-\frac{z^2}{2}\right) \leq \delta \Rightarrow \frac{z^2}{2} \geq \ln\left(\frac{1}{\delta}\right) \Rightarrow z \geq \sqrt{2\ln\left(\frac{1}{\delta}\right)}$$

Substitute $z = \frac{\varepsilon\sigma}{|d|} - \frac{|d|}{2\sigma}$:

$$\frac{\varepsilon\sigma}{|d|} - \frac{|d|}{2\sigma} \geq \sqrt{2\ln\left(\frac{1}{\delta}\right)}$$

Drop the second term for simplification:

$$\frac{\varepsilon\sigma}{|d|} \geq \sqrt{2\ln\left(\frac{1}{\delta}\right)} \Rightarrow \sigma \geq \frac{|d| \cdot \sqrt{2\ln\left(\frac{1}{\delta}\right)}}{\varepsilon}$$

In this expression, δ is empirically adjusted to $\delta/1.25$ to make the bound safer. So, the final expression looks like the following:

$$\boxed{\sigma \geq \frac{|d| \cdot \sqrt{2\ln\left(\frac{1.25}{\delta}\right)}}{\varepsilon}}$$

For Multi-Step Updates (Using RDP Moment)

As per the setting, the original gradient distribution is perturbed with zero mean and σ^2 variance. Now we have to compute the Rényi divergence (RD) between the original distribution and the perturbed distribution.

We see that we have two distributions having the same variance but different means. This is because the noise-adding procedure keeps the standard deviation of the noisy data the same, but the mean changes slightly.

The Rényi divergence of order $\alpha > 1$ between two distributions P and Q is defined

as:

$$D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \int p(x)^\alpha q(x)^{1-\alpha} dx$$

Now, P and Q are Gaussian distributions with means μ_1 and μ_2, and variances $\sigma_1^2 = \sigma_2^2 = \sigma^2$.

Therefore, after plugging in the Gaussian densities of $p(x)$ and $q(x)$, we get:

$$\begin{aligned} D_\alpha(P\|Q) &= \frac{1}{\alpha - 1} \log \left(\exp \left(-\frac{\alpha(1-\alpha)(\mu_1 - \mu_2)^2}{2\sigma^2} \right) \right) \\ &= \frac{\alpha(\mu_1 - \mu_2)^2}{2\sigma^2} \\ &= \frac{\alpha \Delta^2}{2\sigma^2}, \quad \text{where } \Delta = \|\mu_1 - \mu_2\|_2 \end{aligned}$$

The Δ is the sensitivity (max difference between points on neighbouring datasets). This is determined by the point at which the gradient is clipped.

As Δ is set to 1, we obtain:

$$D_\alpha(P\|Q) = \frac{\alpha}{2\sigma^2}$$

This divergence is computed per individual update. Now, in mini-batch SGD and ADAM training, the probability of choosing a batch is q. So, we will have following relation

$$D_\alpha^{\text{batch}} \leq \frac{q^2 \alpha}{2\sigma^2}$$

For training over T epochs, the total Rényi divergence accumulates additively, giving:

$$D_\alpha^{\text{total}} \leq T \cdot \frac{q^2 \alpha}{2\sigma^2} \tag{1}$$

Now, suppose two datasets D and D' differ by only one sample. The privacy loss at output o is defined as:

$$\text{Privacy Loss at } o = \log \left(\frac{Pr[\mathcal{M}(D) = o]}{Pr[\mathcal{M}(D') = o]} \right)$$

In a more general form, this is written as:

$$L(o) = \log \frac{P(o)}{Q(o)}$$

The moment generating function (MGF) of the privacy loss random variable $L(o)$ at a point λ is:

$$\mathbb{E}_{o\sim Q}\left[e^{\lambda L(o)}\right]$$

In Rényi Differential Privacy (RDP) of order α, we set $\lambda = \alpha - 1$. So we have:

$$\mathbb{E}_{o\sim Q}\left[e^{(\alpha-1)L(o)}\right] \leq e^{(\alpha-1)D_\alpha(P\|Q)}$$

Applying Markov's inequality, we have,

$$\Pr[L(o) \geq \varepsilon] \leq \frac{\mathbb{E}_{o\sim Q}\left[e^{(\alpha-1)L(o)}\right]}{e^{(\alpha-1)\varepsilon}} \leq \frac{e^{(\alpha-1)D_\alpha}}{e^{(\alpha-1)\varepsilon}}$$

If we call tail probability δ, we obtain:

$$\delta \leq \exp\left((\alpha-1)(D_\alpha - \varepsilon)\right)$$

Taking logarithms and rearranging the terms gives:

$$\varepsilon \geq D_\alpha + \frac{\log(1/\delta)}{\alpha - 1} \tag{2}$$

Thus, the mechanism satisfies (ε, δ)-DP for any $\alpha > 1$, where:

$$\varepsilon = D_\alpha + \frac{\log(1/\delta)}{\alpha - 1}$$

From Equation (1), we earlier derived the total Rényi divergence after T epochs with batch selection probability q as:

$$D_\alpha = \frac{Tq^2\alpha}{2\sigma^2}$$

So,

$$\epsilon = \frac{Tq^2\alpha}{2\sigma^2} + \frac{\log(1/\delta)}{\alpha - 1}$$

3.4. Datasets:

The first skin image analysis challenge globally, organized by the International Skin Imaging Collaboration (ISIC) that has established the largest public archive of dermoscopic skin images. The challenge was hosted in 2018 at the Medical Image Computing and Computer Assisted Intervention (MICCAI) conference in Granada, Spain. The dataset included over 12,500 images across 3 tasks. The first paper addressing Task 3 (Disease Classification) of the ISIC 2018 Challenge was submitted by Hardie et al [8]. from the University of Dayton.

In this research, will used the challenge dataset ISIC 2018(Task 3) [10], Skin Lesion Analysis. In training set, there are a total of 10015 skin lesion images from seven skin diseases- Melanoma (1113), Melanocytic nevus (6705), Basal cell carcinoma (514), Actinic keratosis (327), Benign keratosis (1099), Dermatofibroma (115) and Vascular (142). The validation dataset consists of 193 images. The test dataset consists of 1512 images. Sample images from all seven lesion types are shown in Figure 3.1

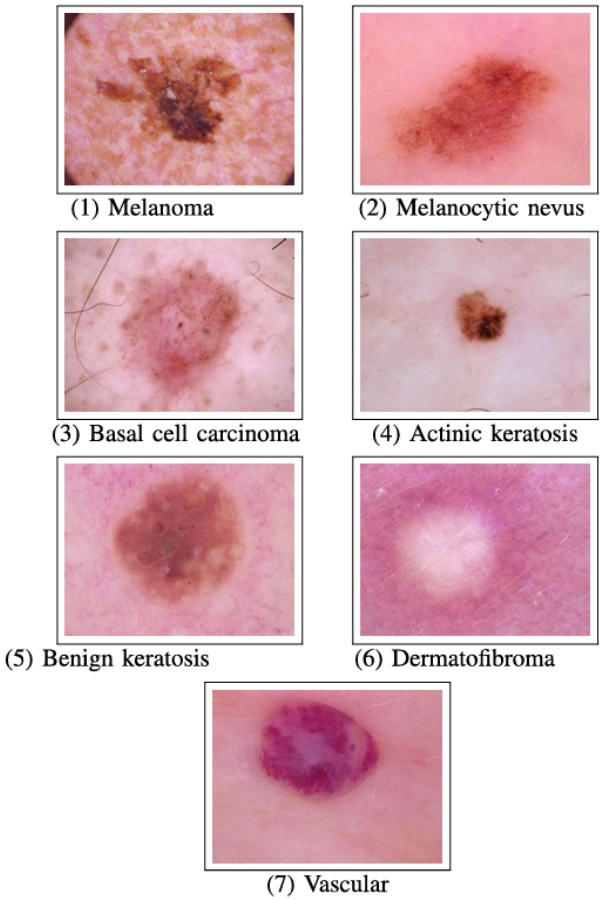


Figure 3.1: Images of different skin lesion

Chapter 4

Experiments and Results

Result and Discussion

Here relation between noise and privacy budget is

$$\sigma = \frac{\sqrt{2\log\left(\frac{1.25}{\delta}\right)}}{\epsilon}$$

Let $\delta = 10^{-5}$.

4.1. SGD and DP-SGD with ResNet50 model

Here we used the custom Convolution Neural Network(CNN) ResNet50 model with Optimizer SGD.

We use Pretrained ResNet50 model.

Use Dense(28, ReLU), fully connected layer with 28 units and ReLU activation and applies dropout(0.4) i.e. 40% to reduce overfitting. Final layer is Dense(7, Softmax) for multi-class classification with 7 output classes.

The hyperparameter setting of our SGD optimizers are:
learning rate = 0.0001
batch size = 20,
number of epoch = 9.

We run this program in NVIDIA gpu and the graph of the training loss vs validation loss is bellow

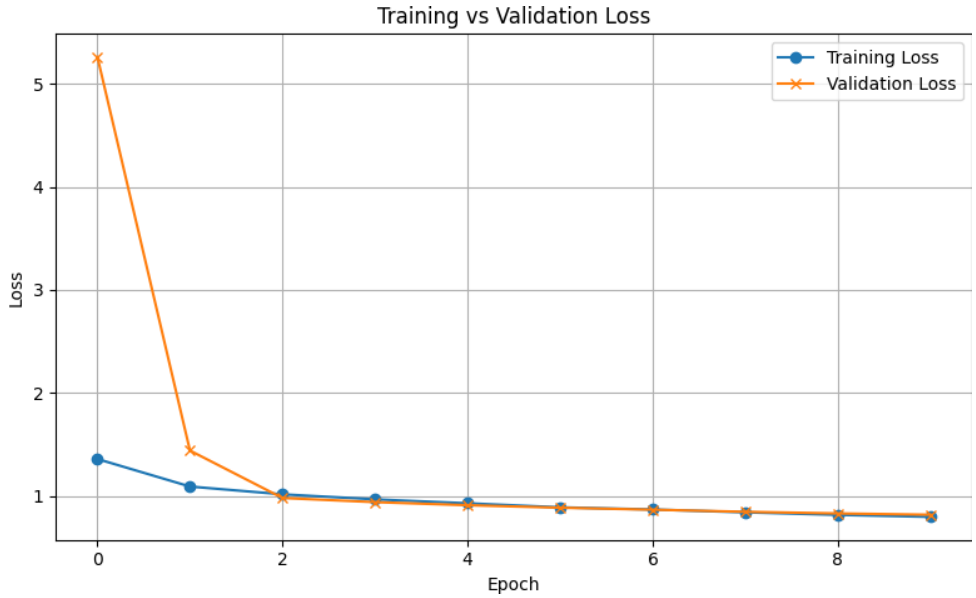


Figure 4.1: Training loss vs Validation loss

In the epoch 9 training loss and validation loss is same, also minimum. So, we run the model upto 9 epoch.

Final Testing accuracy is 64.68%.

For DP-SGD, we use same hyperparameter setting and same CNN architecture and different values of noise σ , get the value of ϵ

Accuracy with DP-SGD in different values of ϵ is given below:

Values of ϵ	**Accuracy (using DP-SGD)**
0.59812	58.60%
0.58371	58.40%
0.56997	58.86%
0.55687	57.94%
0.54436	58.40%

4.2. SGD and DP-SGD with ResNet18 model

In this work, we employed a custom deep Convolutional Neural Network (CNN) based on the pretrained ResNet18 architecture, using the Stochastic Gradient Descent (SGD) optimizer.

The ResNet18 model was used as a feature extractor, followed by custom fully connected layer with 28 neurons and ReLU activation enables the model to capture non-linear

combinations of features extracted by the convolutional base. To mitigate overfitting, a Dropout layer with a rate of 0.2 is applied immediately after, randomly deactivating 20% of the neurons during training.

The final classification layer is a Dense layer with 7 classes and softmax activation, suitable for multi-class classification involving 7 output classes.

The hyperparameter settings used for the SGD optimizer were as follows:

Learning rate: 0.0001

Batch size: 20

Number of epochs: 20

We run this program in NVIDIA GPU and the graph of training loss versus validation loss is shown below.

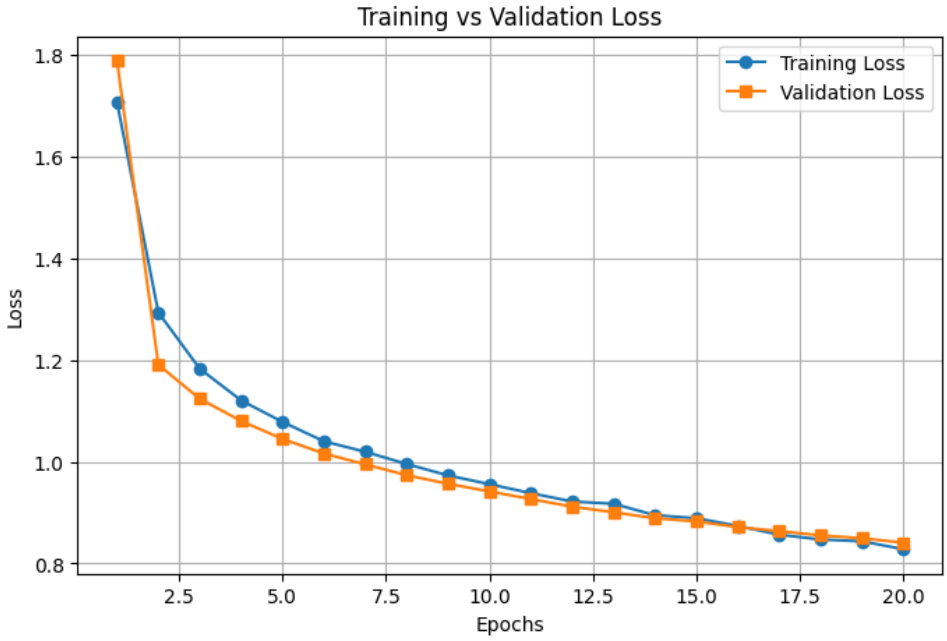


Figure 4.2: Training loss vs Validation loss

As observed, by the 20th epoch, both training and validation losses converge and reach their minimum values. Therefore, training was terminated after 20 epochs Finally, we get the testing accuracy is 66.60%.

For training with Differentially Private Stochastic Gradient Descent (DP-SGD), we employed the same CNN architecture ResNet50 and retained the same hyperparameter settings used in the standard SGD training. The only variation introduced was in the value of the noise multiplier σ, which directly influences the differential privacy guarantee quantified by ϵ.

Using a fixed value of δ, the corresponding values of ϵ were computed for each setting of σ using the standard privacy accounting mechanisms. The model was trained under these conditions and the resulting classification accuracies were recorded.

The table below presents the classification accuracy achieved under various privacy budgets ϵ:

Accuracy with DP-SGD is given below:

Values of ϵ	**Accuracy (using DP-SGD)**
0.59812	59.79%
0.58371	59.72%
0.56997	60.12%
0.55687	60.12%
0.54436	60.05%

4.3. ADAM and DP-ADAM with ResNet50 model

We employed a custom deep Convolutional Neural Network (CNN) based on the pre-trained ResNet50 architecture, using the Adaptive Moment Estimation (ADAM) optimizer. The ResNet50 model was used as a feature extractor, followed by custom fully connected layer with 28 neurons and ReLU activation enables the model to capture non-linear combinations of features extracted by the convolutional base. To mitigate overfitting, a Dropout layer with a rate of 0.1 is applied immediately after deactivating 10% of the less important neurons during training. The final classification layer is a Dense layer with 7 units and softmax activation, suitable for multi-class classification involving 7 output classes.

The hyperparameter settings used for the ADAM optimizer were as follows:

Learning rate: 0.0001

Batch size: 20

Number of epochs: 10

The graph of training loss versus validation loss is shown below.

As observed, by the 10th epoch, both training and validation losses converge and reach their minimum values. Therefore, training was terminated after 10 epochs Finally, we get the testing accuracy with ADAM is 65.41%.

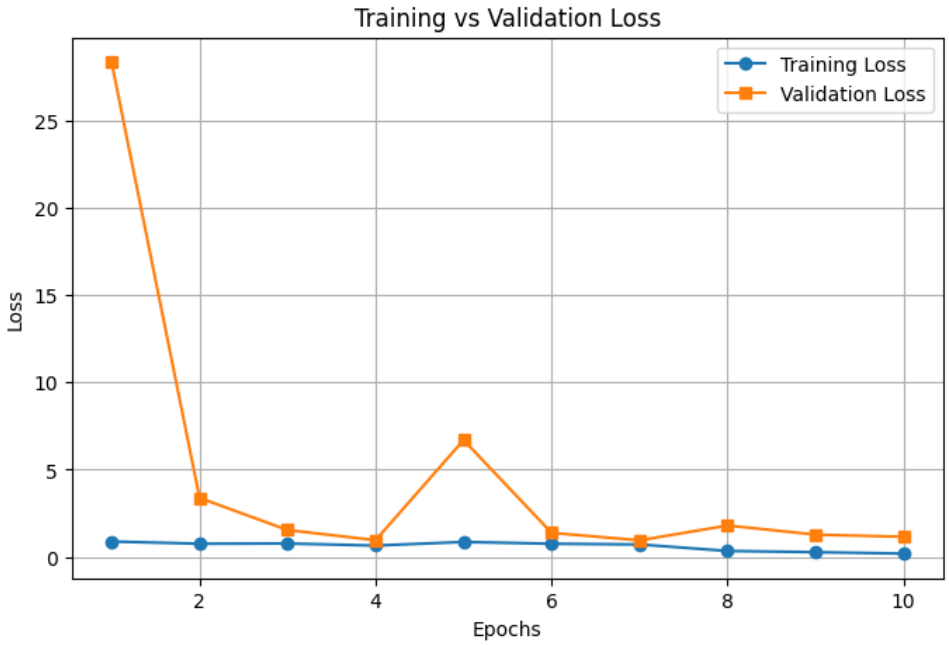


Figure 4.3: Training loss vs Validation loss

For training with Differentially Private Adaptive Moment Estimation (DP-ADAM), we employed the same CNN architecture (based on ResNet50) and retained the same hyperparameter settings used in the standard ADAM training. The only variation introduced was in the value of the noise multiplier σ, which directly influences the differential privacy guarantee quantified by ϵ.

Using a fixed value of $\delta = 10^{-}5$, the corresponding values of ϵ were computed for each setting of σ using the standard privacy accounting mechanisms. The model was trained under these conditions and the resulting classification test accuracies were recorded.

The table below presents the classification accuracy achieved under various privacy budgets ϵ: Test Accuracy with DP-ADAM is given below:

Values of ϵ	**Accuracy (using DP-ADAM)**
0.59812	59.83%
0.58371	59.91%
0.56997	60.12%
0.55687	60.05%
0.54436	60.12%

4.4. ADAM and DP-ADAM with ResNet18 model

We utilized a custom deep Convolutional Neural Network (CNN) built upon the pre-trained ResNet50 architecture, optimized using the Adaptive Moment Estimation (Adam) algorithm. The ResNet50 model served as a powerful feature extractor, followed by a custom fully connected layer comprising 28 neurons with ReLU activation, enabling the network to capture complex, non-linear feature combinations derived from the convolutional base.

To reduce the risk of overfitting, a Dropout layer with a rate of 0.2 was applied, which randomly deactivates approximately 20% of less significant neurons during training. The final classification layer consists of 7 units with softmax activation, making it suitable for multi-class classification involving seven output categories.

The hyperparameters for Adam optimization were configured as follows:

Learning rate: 0.0001

Batch size: 20

Number of epochs: 7

We run this program in NVIDIA GPU and the training and validation loss trends are illustrated in the figure below:

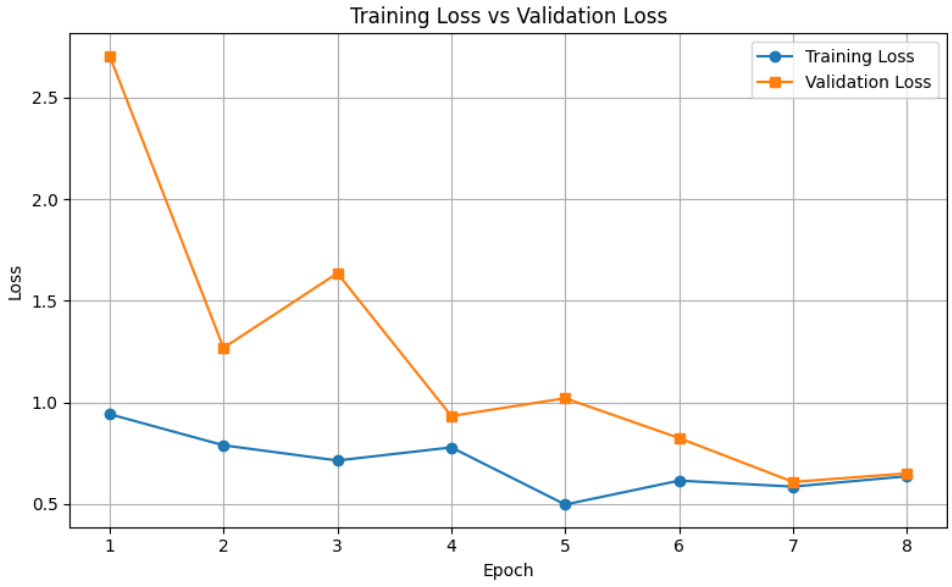


Figure 4.4: Training loss vs Validation loss

As shown, both training and validation losses converge and reach their minimum around the 7th epoch. Consequently, model training was terminated at epoch 7, and the

final test accuracy achieved using Adam is 63.43

For training with Differentially Private Adam (DP-Adam), the same CNN architecture and hyperparameter settings were employed. The only change introduced was in the value of the noise multiplier σ, which governs the level of differential privacy by controlling the magnitude of added noise.

Using a fixed privacy parameter $\delta = 10^{-}5$, the corresponding values of the privacy budget ϵ were computed for different σ values using a standard privacy accounting mechanism. The model was trained under these settings, and the resulting test accuracies were recorded.

The table below summarizes the classification accuracy achieved for various levels of privacy budget ϵ.

Test Accuracy with DP-ADAM is given below:

Values of ϵ	**Accuracy (using DP-ADAM)**
0.59812	60.12%
0.58371	60.12%
0.56997	60.12%
0.55687	60.10%
0.54436	60.05%

4.5. Observation

In our privacy-preserving setting, we observe that the performance degradation caused by the integration of Differential Privacy (DP) is relatively small. Among the four evaluated models, **ResNet18 combined with the DP-Adam** optimizer gives the best trade-off between privacy protection and model accuracy, demonstrating its effectiveness in balancing utility and confidentiality.

Chapter 5

Conclusion and Future Works

The training dataset suffers from data imbalance, which adversely impacts model performance. To tackle this challenge, we plan to implement strategies such as class weighting, oversampling and data augmentation in the later stages of our study to enhance the model's fairness and generalization ability.

While our model demonstrates promising predictive performance, interpretability remains a critical concern, especially in privacy-sensitive applications. In future work, we aim to incorporate model-agnostic interpretability techniques to better understand feature importance and decision rationale, ultimately enhancing transparency and trust in the system.

Next, we will perform a comprehensive security analysis of the trained model against standard privacy attacks, such as membership inference, model extraction, and reconstruction attacks. By simulating these attacks, we will evaluate how much information or data-leakage our model inadvertently exposes and quantify the extent of privacy risk. This analysis will help assess the real-world applicability of the model and guide the implementation of stronger privacy-preserving mechanisms.

Bibliography

[1] The European General Data Protection Regulation (GDPR). https://gdpr-info.eu/ (2019), accessed: Mar. 1, 2019

[2] Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. p. 308–318. CCS'16, ACM (Oct 2016), https://arxiv.org/pdf/1607.00133

[3] Centers for Disease Control and Prevention (CDC): Health insurance portability and accountability act of 1996 (hipaa). https://www.cdc.gov/phlp/php/resources/health-insurance-portability-and-accountability-act-of-1996-hipaa.html (1996)

[4] Chicco, D., Sadowski, P., Baldi, P.: Deep autoencoder neural networks for gene ontology annotation predictions. In: Proceedings of the ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM BCB). pp. 533–540 (2014)

[5] Ching, T., Himmelstein, D.S., Beaulieu-Jones, B.K., Kalinin, A.A., Do, B.T., Way, G.P., Ferrero, E., Agapow, P.M., Zietz, M., Hoffman, M.M., Xie, W., Rosen, G.L., Lengerich, B.J., Israeli, J., Lanchantin, J., Woloszynek, S., Carpenter, A.E., Shrikumar, A., Xu, J., Cofer, E.M., Lavender, C.A., Turaga, A., Alexandari, A., Lu, Z., Harris, D.J., DeCaprio, D., Qi, Y., Kundaje, A., Peng, Y., Wiley, L.K., Segler, M.H., Boca, S.M., Swamidass, S.E., Huang, A., Greene, B.A.: Opportunities and obstacles for deep learning in biology and medicine. Journal of the Royal Society Interface **15**(141) (2018). https://doi.org/10.1098/rsif.2017.0387

[6] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) Theory of Cryptography(Lecture Notes in Computer Science), vol. 3876, pp. 265–284. Springer, Berlin, Germany (2006)

[7] Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science **9**(3–4), 211–407 (2014), https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf

[8] Hardie, R.C., Ali, R., Silva, M.S.D., Kebede, T.M.: Skin lesion segmentation and classification for isic 2018 using traditional classifiers with hand-crafted features (2018), https://arxiv.org/abs/1807.07001

[9] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015), https://arxiv.org/abs/1512.03385

[10] (ISIC), I.S.I.C.: Isic 2018 (2018), https://challenge.isic-archive.com/data/#2018

[11] Liu, S., Wang, Z., Chen, Y., Lei, Q.: Data reconstruction attacks and defenses: A systematic evaluation (2025), https://arxiv.org/abs/2402.09478

[12] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). vol. 54, pp. 1273–1282. PMLR (2017)

[13] Perozzi, B., Al-Rfou, R., Skiena, S.: "deepwalk: Online learning of social representations". In: Proceeding International Conference on Knowledge Discovery Data Mining (KDD). pp. 701–710 (2014)

[14] Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models (2017), https://arxiv.org/abs/1610.05820

[15] Tramèr, F., Zhang, F., Juels, A., Reiter, M.K., Ristenpart, T.: Stealing machine learning models via prediction apis (2016), https://arxiv.org/abs/1609.02943