Disease Classification on Encrypted Retina Scan Images using Neural Network

Anri Abrahamyan,
Elina Israyelyan,
Hovhannes Manushyan,

Supervisor: Varduhi Yeghiazaryan

Submitted for the
degree of

BS Data Science

College of Science and Engineering

American University of Armenia

2023

Abstract

As more sensitive medical data is being utilized for machine learning applications, ensuring the privacy and security of such data is becoming increasingly important. This paper proposes a privacy-preserving approach for disease detection on retinal scan images using homomorphic encryption. We explore the use of convolutional neural networks and vision transformer models for this task and highlight the need for custom approximation functions for certain activations like ReLU, GeLU, and Softmax. Our experiments show promising results in terms of accuracy and privacy preservation and demonstrate the feasibility of using homomorphic encryption for medical image analysis. We propose a vision transformer architecture with an 83% accuracy on the testing dataset and obtain an approximation function for layer normalization operation, getting us one step closer to performing fully encrypted inference. Overall, our approach offers a potential progress for protecting sensitive medical data while enabling the advancement of machine learning in healthcare.

Table of Contents

Chapter 1

Introduction

Third-party computation sevices are gaining popularity, and computationally heavy procedures such as machine learning (ML) inference are a large part of this movement. Furthermore, large companies in various fields are already implementing the conversion of their daily processes to automated pipelines where manual tasks are substituted by ML technologies. In the medical sphere such automated ML-based approaches include the procedures of treatment planning, clinical studies, disease detection and prevention [17, 48, 49, 33, 24]. For example, the application of segmentation techniques to computed tomography (CT) and magnetic resonance (MR) images can facilitate the analysis of anatomical structures, enable the identification of regions of interest such as tumors, lesions, and other anomalies, allow the measurement of tissue volume to assess changes in tumor size, and facilitate radiation dose calculations and other important processes [46]. ML techniques are also widely used for tasks in neurology, cardiology, psychiatry, [47, 36]. The need for ML-based approaches is big, and companies, research labs, hospitals, vendors work together to find optimal solutions for medical imaging using deep learning [36, 49].

Besides the advancements in the field, the medical spheres are concerned with data privacy, and these limitations prevent people in the field from using cloud services [47, 36, 5]. Medical imaging involves interaction with sensitive data such as CT scans, X-rays, retina scan images, etc. This data is usually accompanied by metadata for identifying the person scanned. Recent advances in biometrics allow identifying the person using solely physical characteristics such as retinal scan images. Retinal

scan, a biometric method utilizing a low-intensity light source, is capable of creating a distinctive map of the individual's retinal pattern, which is employed effectively for the purpose of human recognition [18, 44, 30, 12]. Furthermore, the retinal pattern of the subject undergoes little change during their lifetime compared to other forms of scans, such as fingerprints, and it is not exposed to the threats of the external environment. Furthermore, the pattern is unique, so two people can't have the same retinal vascular pattern [4].

The issue of data privacy is faced by organizations leveraging sensitive data such as gender, ethnicity, and sexual and political orientation [19]. For this reason, these organizations must protect data privacy through the entire data life cycle. These requirements present a problem for machine learning applications, which need to extract actionable information from the data while following the data privacy requirements.

For these reasons, privacy-preserving methods for machine learning are developed. One of the prominent approaches is Federated learning [19]. The method implies of participants privately training their models and only sharing final result with other participants. This again presents a privacy concern since local parameters are shared. Another method, which is used in this project, is homomorphic encryption (all homomorphic encryption papers). With this approach ML-based models result in more secure computations. The current approach enables encrypted and privacy-preserved data flow and does not expose anything to any party involved in the process of ML inference.

This capstone project tackles the problem of using ML-based disease detection on retinal scan images using homomorphically encrypted inputs in order to preserve the privacy of the patients.

Chapter 2

Literature review

2.1 Retinal Fundus Scan Classification

2.1.1 Datasets and preprocessing pipelines

We consider one of the tasks from Retinal Image Analysis for multi-Disease Detection Challenge (RIADD) competition using the retinal fundus multi-disease image dataset (RFMiD) [39]. This dataset was created to enable the development of automated disease classification tools for retinal fundus scans. It is a collection of 3200 retinal fundus scans. The dataset has two versions that differ in the number of disease classes per retinal scan available in the competition website [1]. The images were captured using three different cameras. The dataset is divided into three parts where 60% of the data is designated for training, 20% for evaluation and the other 20% for testing. Therefore, this dataset is an existing benchmark for model performance both on plain and privacy-sensitive data. In chapter 3 we compare our results to the state-of-the-art research performed on this task.

Recently, the RIADD team released an extension of this dataset with additional images and an almost identical setup for collecting retinal fundus scan images [40]. We don't use this additional data to be able to compare our results to the original research from the RIADD competition.

The preprocessing pipeline that we use is the preprocessing pipeline of the winning team of the RIADD competition [1]. The Chapter 3 discusses this pipeline in detail.

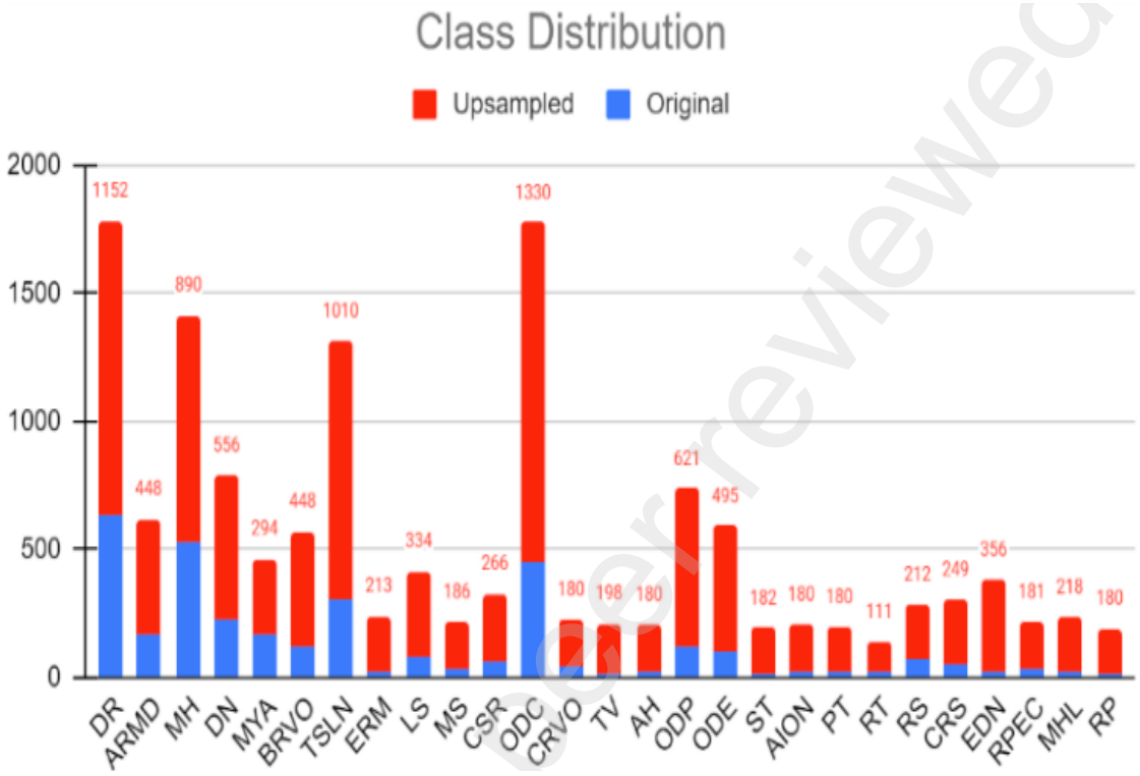

Figure 2.1: RFMiD class distribution after up-sampling by Khan et al [27, 39]

.

The first runner-up team employed a similar pipeline, with the main components being black edge removal and data augmentation during the training phase. They used a random horizontal flip and a random rotation for data augmentation [1].

The second runner-up team employed a similar black edge removal and data augmentation pipeline. For data augmentation, they used RandomResizedCrop, Cutout, CoarseDropout, Horizontal Flip, ShiftScaleRotate, HueSaturationValue, RandomBrightnessContrast and CLAHE [1].

Khan et al. used the dataset with 27 classes [27]. Their preprocessing pipeline involved image transformations in the form of contrast alteration, resizing and cropping. They employed image augmentation with dataset up-sampling to triple the dataset size and guarantee that each disease class had at least 100 samples. The resulting distribution can be seen on Figure 2.1. This approach of up-sampling uses the available specific disease information in the dataset.

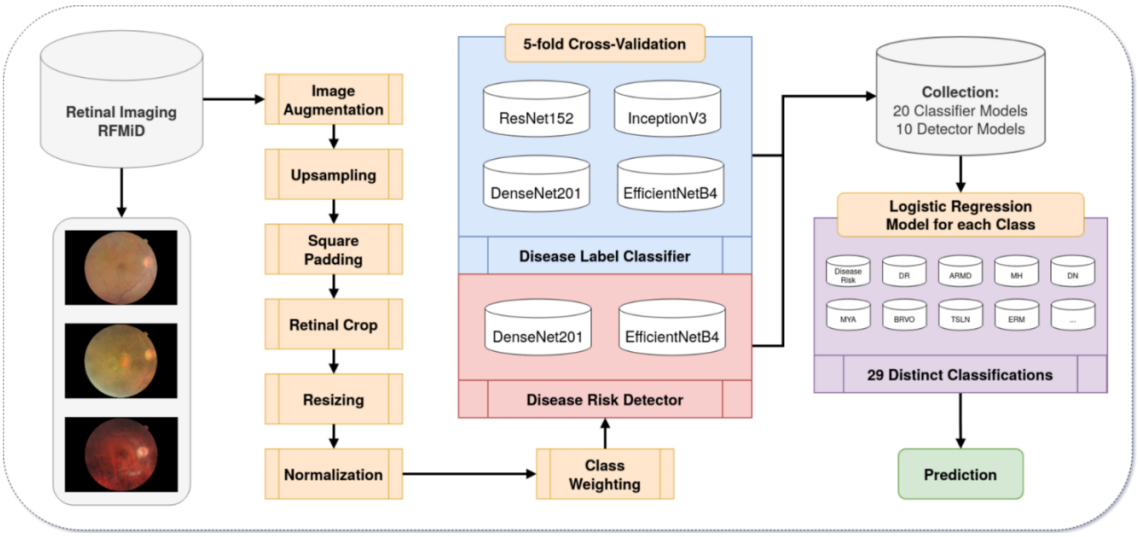

Figure 2.2: RFMiD classification pipeline presented by Müller et al [37, 39]

.

Kumar et al. used a simpler pipeline for preprocessing [29]. They used rotation and flipping for data augmentation. They didn't specify the details of the up-sampling procedure, but it was used to almost double the number of available scans. They also used Z-score normalization and image resizing for their scans.

Müller et al. utilized the pipeline shown in the following Figure 2.2. The image data augmentation was applied to have at least 100 samples of each disease class, almost doubling the dataset size. The augmentation pipeline consisted of rotation, flipping, and altering in brightness, saturation, contrast, and hue. The preprocessing pipeline included image cropping, square padding, and Z-score normalization.

Rodriguez et al. modified these approaches by first combining the RFMiD dataset with other existing datasets for similar task [43]. These datasets are automated retinal image analysis (ARIA) and structured analysis of the retina (STARE) [54, 22]. In their data preprocessing procedure, they used a preprocessing technique of combining multiple low-frequency classes into one class. In their case, it is called 'OTHER'. Furthermore, they employed a pipeline similar to the rest of the research with data augmentation and random up-sampling [1].

2.1.2 ML model solutions

This section covers model architectures applied in the literature to tackle the retinal image scan disease classification task. We also discuss techniques used during model training to yield better results. It should be noted that we do not focus on architecture choices in the existing literature, which do not satisfy constraints imposed by the encryption method, as defined in Chapter 3.

One of the common approaches used in the literature is the reduction of the number of disease classes in the RFMiD dataset [1, 43]. This approach combined multiple diseases, usually with similar ranks in frequency, into one class. Therefore, the number of necessary prediction models decreased.

In addition, cross-validation was often used in the model training pipeline for hyper-parameter tuning [37, 27, 1]. These hyper-parameters included the learning rate, class weights, weighted focal loss function parameters, and others. The image size was fixed in accordance with the chosen neural network architecture.

There were multiple convolutional neural network architectures used in the literature for retinal scan classification. These included various EfficientNet architectures [29, 27, 43, 52, 53]. In addition, researchers experimented with ResNet [23], DenseNet [25], and InceptionNet [50] architectures [29, 27, 43]. Furthermore, Rodriguez et al. implemented the C-Tran transformer architecture for this classification task [43]. These architectures utilized transfer learning because they were pre-trained on the ImageNet dataset [13].

Finally, others experimented with the model optimizer and the loss functions. Müller et al. used a focal loss to tackle the class imbalance [37]. Also, the second runner-up team of the RIADD competition used a ranger optimizer, a combination of the Adam and lookahead optimizers [1].

Table 2.1: Performance evaluations on the RFMiD dataset [1]

Authors	Accuracy	AUC
Khan et al. [27]	0.96	NaN
RIADD winner [1]	0.88	NaN
Rodriguez et al. [43]	NaN	0.96
Kumar et al. [29]	NaN	0.98

2.1.3 Existing results on the RFMiD dataset

Model performance on the RFMiD dataset is evaluated using three main metrics. These metrics are accuracy, the area under the receiver operating characteristic (AUROC), and mean average precision (MAP). Also, there are papers reporting their results using the area under the ROC Curve (AUC) score [43, 27]. Table presents 2.1 the currently available performance results in the existing literature using the AUC metric since it is the most common metric used for reporting results. Note that we did not include results from research that was not evaluated on the test set of RFMid [39].

2.2 Encryption in Machine Learning

We examine a range of papers that explore various privacy-preserving ML (PPML) techniques, with a particular focus on their effectiveness and practicality in real-world settings. PPML solutions include different techniques, such as federated learning (FL), homomorphic encryption (HE), or client-side ML serving. In general, techniques for ensuring privacy compliance in machine learning models can be categorized into three main groups: anonymization, perturbation, and distributed protocols. Anonymization techniques aim to protect the privacy of individuals in a dataset by obscuring personally identifiable information while maintaining the utility of the data. Perturbation techniques add noise to the data, machine learning algorithm, or learned model to corrupt the disclosed information. Furthermore, perturbation techniques ensure data privacy and security by utilizing distributed protocols to distribute the data across different entities. An example of such an approach is fed-

erated learning [19]. While all these have advantages and disadvantages, they are already experimented on in empirical cases, with a huge amount of research effort committed in the recent decade. Adnan et al. report federated learning techniques used on lung cancer images [2]. They show that federated learning helps improve the final model's accuracy. FL produces MRI segmentation masks that perform better or are comparable to models trained on-premises. This approach helps to determine hospital triage for the level of care and oxygen requirement in patients with COVID-19, according to Darzidehkalani et al. giving an additional boost to FL in radiology [10, 11]. Another paper presents a use case of FL for medical image analysis. It examines the advantages and improvements in predictions and execution times compared to centralized approaches and addresses issues related to varying numbers of clients and intermittent clients [15]. However, it is worth mentioning that in the case of FL, we refer to private model training rather than private inference. Hence FL solutions do not solve the problem of privacy-preserving inference, which we tackle.

For private inference, homomorphic encryption (HE) is one of the field's most advanced and common techniques. The first fully homomorphic encryption (FHE) scheme was introduced by Craig Gentry in 2009 [20]. HE can guarantee the privacy of an individual's data, particularly when a third-party service provider offers memory and computing resources. However, using HE significantly increases the computational demands and limits the potential options for the neural network architecture.

In 2011 Lauter et al. showed a proof-of-concept implementation of HE that relied on the "ring learning with errors" (Ring LWE) problem [31]. The system was very efficient, and it had reasonably short ciphertexts on which the common modern implementations of HE are constructed [45, 9, 35].

Dowlin et al. introduce CryptoNets on the modified national institute of standards and technology database (MNIST) optical character recognition tasks. CryptoNets achieve 99% accuracy and can make more than 51000 predictions per hour on encrypted images using HE while running on a single PC [21].

Lee et al. showcase an HE-ready model on the ImageNet dataset [14] with 77.52% accuracy, which is very close to the original model accuracy of 78.31% [32].

Jin et al. introduce CareNets, which implement homomorphic encryption on high-resolution images, while prior works proposed approaches for images with size 32×32 [6]. CareNets work on retinal images of sizes 96×96 and 256×256. Jin et al. introduce the first ciphertexts packing method to reduce the size of ciphertexts and enable encryption of bigger matrices.

Mihara et al. show the feasibility and advantages of HE on the Iris dataset using the SEAL library [34, 45]. The same result is shown by Onoufriou et al. by performing time and space complexity analysis on CKKS the [9] HE scheme [38]. Yang et al. emphasize the importance and feasibility of HE in biometric identification feature encryption. They state that HE still faces unsolved issues, such as high computational complexity, low efficiency, and inadequate deployment in the real world. Further research is needed to make HE-related encryption, decryption, and matching processes more efficient and practically implementable [57].

Qi et al. and Kiya et al. show an encrypted inference implementation on vision vransformers [28, 41] using trainable encryption.

In our report, we discuss encrypted inference on vision vransformers but use homomorphic encryption for simplicity and feasibility of implementation. We use the TenSEAL library [3] built on the+ Microsoft SEAL encryption scheme [45]. We discuss the details of HE and the TenSEAL library in Chapter 3.

Chapter 3

Methodology

3.1 Data preprocessing

We apply our encrypted models on the first task of the RFMiD challenge dataset. This is the version of the dataset with fewer disease labels that are used during the competition. Also, our target variable is the column indicating the existence of a disease in a given retinal fundus scan. For reference, this is the first task of the RIADD competition [1].

We perform exploratory data analysis (EDA) on the dataset to better understand the data structure. The main findings indicate that the data needs preprocessing, which is present in all of the research examined in this paper [29]. Given our problem setup, we do not use the extra information in the form of specific disease names. We focus only on the indicator variable which shows the existence of a disease in a given retinal fundus scan. Therefore, our predictions are based only on the scans of the patient's eye retina.

In order to tackle the problems that we discover during our EDA and to improve the model generalization, we perform data preprocessing on the retinal fundus scans. The general pipeline discussed in the literature involves image preprocessing, image augmentation, and upsampling. The pipeline which we use is the preprocessing pipeline of the winning team of the RIADD competition [1]. The augmentations they use are horizontal flip, vertical flip, random brightness and contrast alteration,

median blur, Gaussian noise addition, hue and saturation alteration, and random cutout.

3.2 Proposed encrypted model architecture

Our study proposes a deep learning architecture that is capable of providing a versatile framework across various domains, while satisfying the limitations of homomorphic encryption. To achieve this goal, we opt for the transformer architecture as our primary choice, given its proven success in the field of natural language processing and its potential for applicability across other domains as well.

In particular, our research focuses on an encrypted version of the vision transformer, which is applied to the Retina Disease Classification task. However, it should be noted that the same encryption scheme can be extended to other transformer architectures in domains such as natural language processing.

The vision transformer architecture used in this work is based on the seminal paper by Dosovitskiy et al. [16]. Our selection of this particular architecture is motivated by several factors, including its foundational contribution to the vision transformer architecture and its potential for enhancing the performance of subsequent works. Additionally, the architecture is relatively simple to develop as a minimal reproducible example of an encrypted transformer.

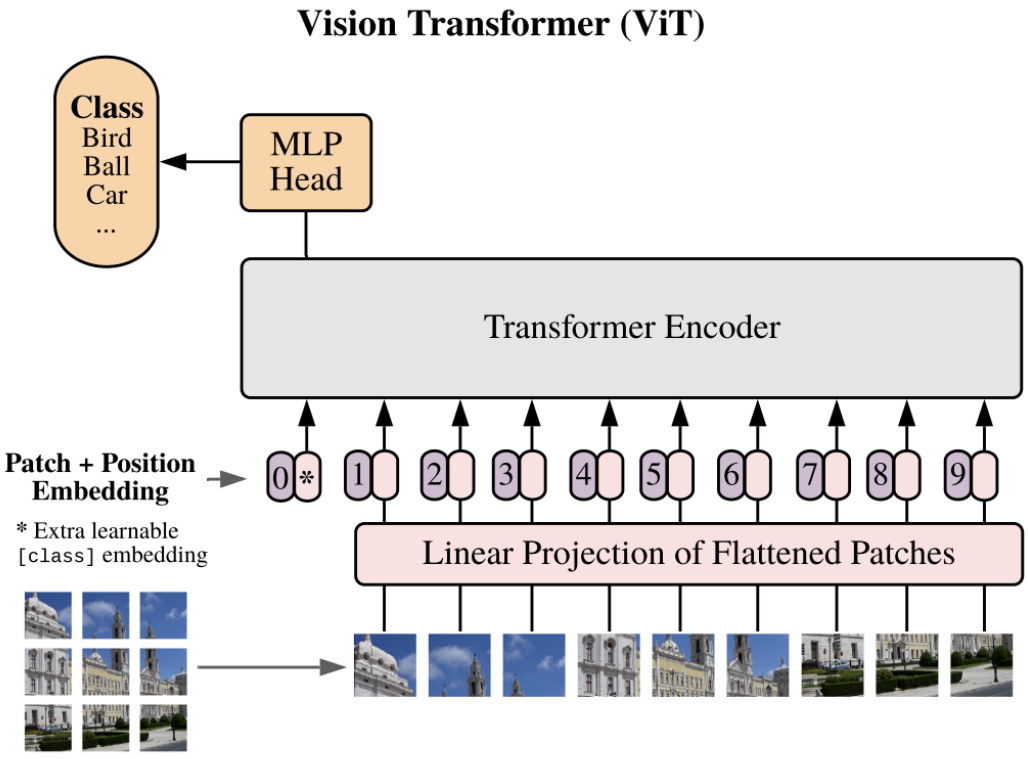

Figure 3.1: Vision Transformer Overview by Dosovitskiy et al.[16]

In this study, we first provide a concise overview of the vision transformer architecture, followed by a detailed discussion of its encrypted variant. The key advantage of the vision transformer architecture, compared to traditional convolutional neural networks (CNNs), arises from its remarkable parallelizability and the potential for polynomial approximation of operations performed by each thread.

This attribute is particularly beneficial in the context of homomorphic encryption, as it enables encryption to be applied to each parallel thread of execution while preserving parallelism and satisfying the intrinsic constraints of encryption. The substantial parallelization in vision transformers is achieved through the division of the input image into a sequence of patches, which are subsequently processed in parallel. The process is illustrated in Figure 3.1. In our particular approach, the input image is segmented into a sequence of 8x8 patches, and each patch is

independently subjected to homomorphic encryption. It is important to highlight that encrypting the entire patch sequence at this stage would not be feasible due to the addition of an encrypted classification token in subsequent stages. Moreover, the polynomial approximations utilized in our approach are only relevant to individual patches.

Following patch-wise encryption, the patches are subjected to a linear embedding layer, which is similar to the architecture proposed by Dosovitskiy et al. The role of embedding layers in transformer architectures is to facilitate the learning of a better representation space for the input patches, thereby streamlining the feature extraction process in downstream layers. A separately encrypted embedded classification token is then prepended to the encrypted patches after the embedding layer.

The multi-headed attention layer plays a crucial role in this phase by extracting similarity features between the classification token and the rest of the tokens. This facilitates the model's ability to pay more attention to the tokens that are more likely to contain disease. In addition, to address the challenge of conveying the location of each pixel to the transformer, we adopt a learnable positional encoding scheme. This approach, as described in "Convolutional Sequence to Sequence Learning" offers a simpler implementation than more common sinusoidal encoding schemes. Furthermore, according to Vaswani et al.[55], there is not a significant difference in performance between the two approaches. After incorporating the positional encoding vectors, we compute the sum between the positional vectors and embedding vectors, which is then fed into the transformer encoder blocks. This step is crucial in enabling the model to leverage the positional information of the tokens and better understand the spatial relationships between them, thereby enhancing its ability to accurately identify disease-containing tokens.

3.2.1 transformer block architecture

The architecture block starts with a layer normalization operation defined in Equation 3.1.

$$y = \frac{x - E[X]}{\sqrt{Var(X) + \epsilon}} \tag{3.1}$$

X is the input vector, which in our case are the embedding vectors for each input patch. This operation normalizes the activations of neurons, which is essential for controlling the gradient scales in transformers. However, not only is the operation's presence important, but also its position in the execution graph. In our case, as the core construct of our transformer block architecture, we choose the Pre-layer normalization method proposed by Xiong et al.[51]

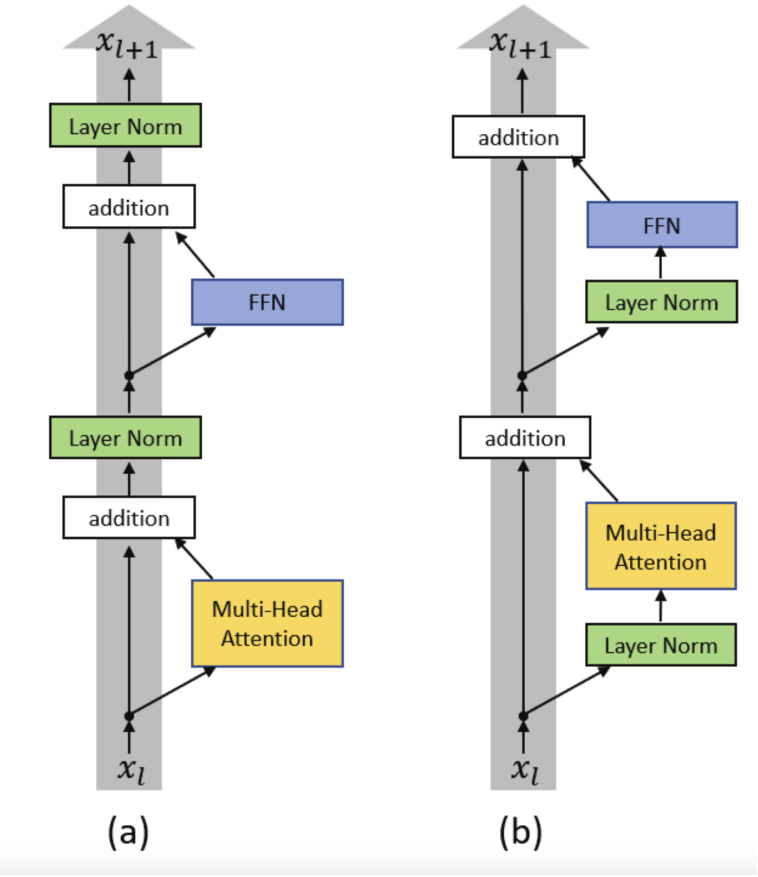

Figure 3.2: Post-Layer vs Pre-layer normalization Overview by Xiong et al.[51]

The difference between the two can be observed in Figure 3.2. Pre-layer normaliz-

ation architecture alleviates the need to have a careful learning rate warm-up stage by reducing the initial gradient scale and variance, thus subsequently allowing for larger learning rates. In addition to layer normalization, the transformer architecture comprises a multi-headed attention (MHA) layer, the characterization of which is necessary to examine its behavior within the context of the encrypted architecture. This layer is a critical element of the transformer architecture, as it models the contextual interactions among the tokens. In the current study, this layer's primary responsibility is to comprehend the correlations between the image patches and how they relate to one another, specifically in terms of the disease classification token. To understand the inner workings of the MHA layer, one has to first get a solid grasp of the attention layer.

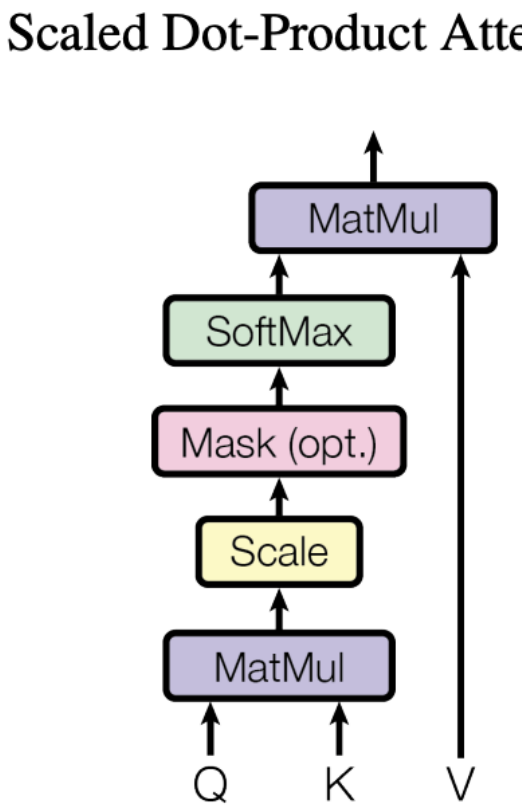

Figure 3.3: Scaled Dot-Product Attention Overview by Vaswani et al. [55]

The input of the attention layer are the query (Q), key (K), and value (V) vectors.

This notion originates from the information retrieval systems, and the attention layer is to compute the similarity between query and key vectors and attribute value to each interaction. In our case, the query and key vectors are two different embeddings of each image patch. Therefore, intuitively, the initial part of the layer computing is the Equation 3.2 where d_k is the dimension of the key embedding vector.

$$SoftMax\left(\frac{Q * K^T}{\sqrt{d_k}}\right) \tag{3.2}$$

This essentially computes the pairwise similarity of input image patches in the context of disease detection. The SoftMax operation is defined in Equation 3.3.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K \tag{3.3}$$

The input z is the matrix containing all patch embeddings with the classification token embedding. SoftMax operation essentially converts the similarity matrix into the range $[0, 1]$ such that the sum of elements in the matrix is 1. Intuitively this acts as a filter over the value matrix highlighting which patches in the input image should be given higher weight in the context of disease classification. The Scaled Dot-Product Attention acts as a the primary block for the MHE layer allowing it to extract features contextually connecting the input patches both in relationship to one another and the given task at hand.

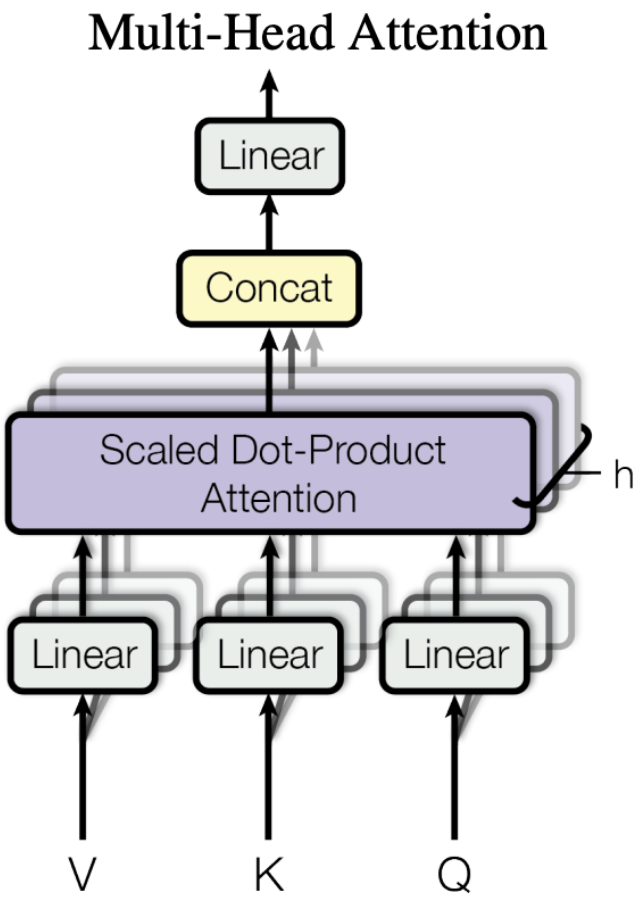

Figure 3.4: Multi-Headed Attention Overview by Vaswani et al. [55]

Based on Figure 3.4, the MHE layer is a concatenation of the output features from multiple Scaled Dot-Product Attention layers followed by a Linear layer. The reason for having concatenation of multiple attention layers is that a single patch might be related to both the task and the other patches in more than one way. Therefore, it is necessary to have multiple layers of attention to capture all the patch interactions.

3.3 Encrypted Inference

3.3.1 Homomorphic encryption

Enabling encrypted inference requires secure and feasible encryption schemes. The encryption scheme converts the original data into an unrecognizable form, which is

hard to decipher. However, for encrypted inference, having the latter is not enough, so an encryption scheme is needed to perform operations on the resulting ciphertext and to retrieve the output of those operations afterward. For this purpose, we use homomorphic encryption (HE). HE is a type of encryption that allows computations to be performed on ciphertext, producing an encrypted result which, when decrypted, is the same as if the computation were performed on plaintext. HE, depending on the used scheme, can support a number of operations like addition, multiplication, and taking exponents. There are two types of HE: fully homomorphic encryption (FHE) and partially homomorphic encryption (PHE). FHE supports multiple available operations like addition, multiplication, exponents, while the PHE can only support one of them. The homomorphic property is defined as follows: for any $m_1 \in \mathbb{R}$, it is possible to compute a ciphertext c_3 such that, upon performing operations on those ciphertexts, the decrypted result is the same as if the operations were done on plain numbers. More precisely, $E^{-1}(f(E(m1))) = f(m_1)$ where $f()$ is any operation, $E()$ is the encrypt function, E^{-1} is correspondingly the decryption function and $m_1 \in \mathbb{R}$.

For HE, we choose the TenSEAL library [3] for its simplicity, the available community, modernity, and flexibility. TenSEAL supports HE schemes such as CKKS and BFV. We choose the CKKS scheme as it supports the encryption of real numbers, unlike the BFV [3, 9].

The main steps behind the CKKS scheme are the following: encoding vectors of real numbers to polynomials and then encrypting the result. The encoding part is essential for good performance of the encryption [26]. First, the vectors of size n are encoded into plaintext polynomials, which are elements of a cyclotomic ring.

Denoting the plaintext polynomials with $\mathbf{Z}_q[X]/(X^N + 1)$ where N denotes the $\frac{N}{2}^{th}$ cyclotomic polynomial, we define the polynomial in such a way that the polynomial $p(x)$ in roots of the $\frac{N}{2}^{th}$ cyclotomic polynomial equals our original vector of size $\frac{N}{2}$. This is the decoding part where we evaluate our polynomial on the roots of the cyclotomic polynomial. However, we also encode, meaning we obtain our polynomial in such a way that decoding results in the original vector. We can see high-level

encryption steps in Figure 3.5. It is worth mentioning that we can only encrypt a vector of size $\frac{N}{2}$ as the polynomial ring $\mathbf{Z}_q[X]/(X^N+1)$ is evaluated on complex roots, and half of the resulting numbers are the conjugates of the other half. Hence, we take a vector of size $\frac{N}{2}$ and expand it by copying the other half with their conjugates [9]. Besides the above-described procedures, other steps are also performed for maintaining precision and rounding techniques for the right projection into the polynomial ring $\mathbf{Z}_q[X]/(X^N+1)$.

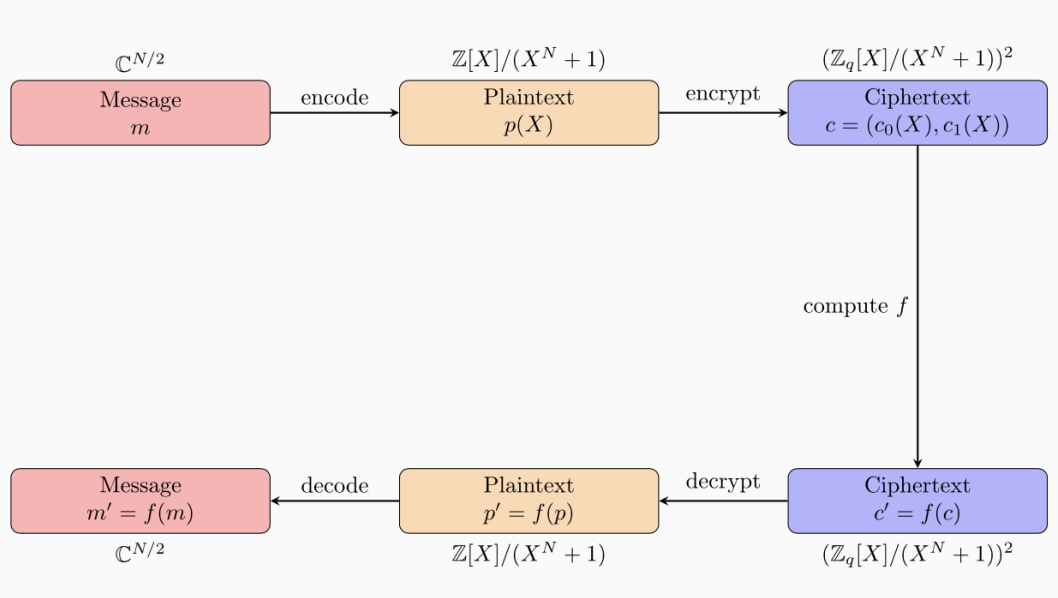

Figure 3.5: High-level view of CKKS [26]

After converting vectors into plaintext polynomials, we create the ciphertexts, i.e., encrypt the message. First, we define public and private keys. In the CKKS scheme, the public key is part of the private key. We only delve a little into the details of key generation, as it is outside our scope. To understand the intuition, we need to introduce the components of a ciphertext and what parts of it might result in errors. So, the ciphertext of message m_1 has two components c_0 and c_1, each containing a generated small error e. When performing the add operation, we only add to the message by $add(ct, m2) = ([ct_0 + \delta * m2]q, ct_1)$ where $m_2 \in \mathbb{R}$ and ct_0 has two components: one that contains the error plus $\delta * m_1$. Hence, m_2 does not add an extra error term. However, for multiplication, we have an emerging extra error as it

is defined in such a way: $mul(ct, m2) = ([ct_0 * m2]q, [ct_1 * m2]q)$, and each element in each ciphertext component is multiplied by m_2. We can imply from the latter that the number of multiplications and the size of the m_2 can affect the size of the error term e. The same applies to ciphertext-to-ciphertext addition and multiplication. Hence, we consider this when defining the number of model architecture layers, the number of nodes on each layer, and the weights of layers.

3.3.2 Results of encrypted inference

We first define a simple model architecture with two linear layers and sigmoid activation functions to perform a sanity check for encrypted inference with torch models. Both linear layers have 16 nodes. Input data is generated randomly from a uniform distribution in the range of [0,1], and the labels are either zero or one, representing if the generated number is bigger than 0.5. The sigmoid activation function is non-linear, but for simplicity's sake, we decrypt the results and then apply the sigmoid function. When running the latter on encrypted numbers in the above-described way, we obtain a mean error of 0.0005. Furthermore, in Figure 3.6, we observe no significant error when encrypting and decrypting a single number, and the mean error is around 0.00002.

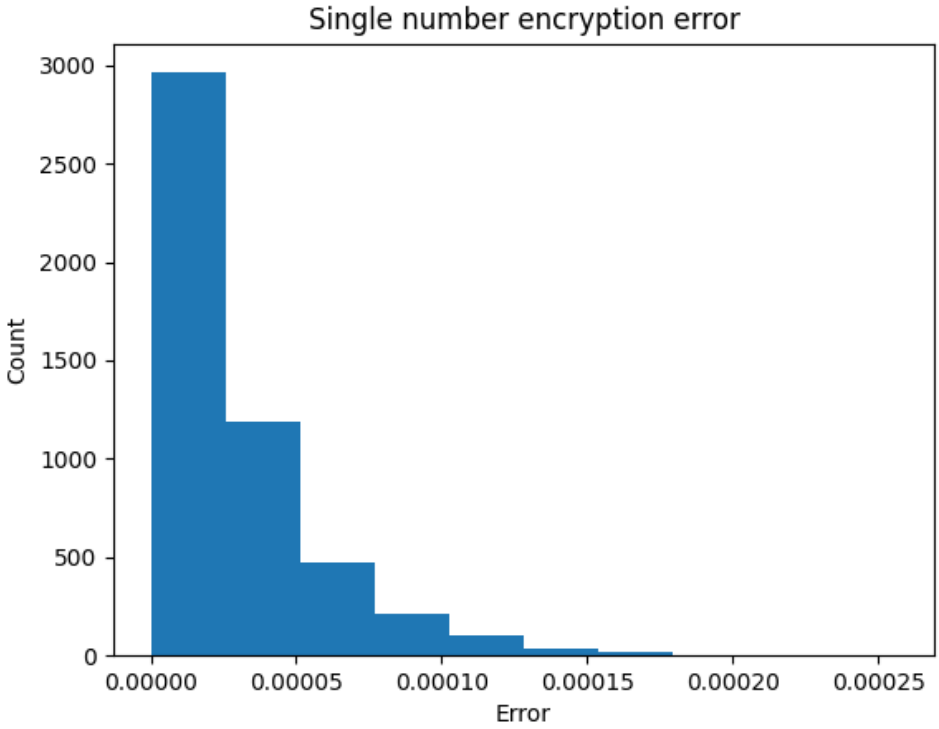

Figure 3.6: Error distribution for encryption using TenSEAL

During this first experiment, we try different parameters for the TenSEAL context. However, we want to emphasize something other than that parameter. We should not ignore the fact that the context parameters, i.e., public and private key parameters and parameters of the encryption scheme, affect the level of security, multiplicative depth (the number of multiplications that can be done), and the size of vectors that can be encrypted (just as described in subsection 3.3.1).

After the sanity check, we can already run our experiments on CNN with the retinal scan images. We use the data from our original dataset of RFMiD [39]. For the first experiment, we use the CNN model with two convolutional layers concatenated and apply two linear layers with sigmoid and ReLU activation functions. There is no convolution operation defined for homomorphic encryption. However, the TenSEAL library supports it by converting the convolution to matrix multiplication by shifting the elements of the original matrix. This means that convolutional neural networks can only be applied once on the first layer as further element shifting inside the

ciphertexts is not defined. In order to make the most out of it, we use two convolutional layers as the first layers and concatenate them in the following steps. This technique enables the use of two or more convolutional layers, concatenating them and extracting more and more features out of images while preserving the privacy of the original data. Hence, this is a way to use multiple convolutional layers. Another challenge is the sigmoid function for which we use an approximation defined in Equation 3.4. The polynomial is a minimax approximation by the Remez algorithm [42] found in the paper by Chen et al. [7]. It is a very good approximation for the sigmoid function in the range of [-5,5].

$$sigmoid(x) = 0.5 + 0.197 * x - 0.004 * x^3 \tag{3.4}$$

We do not discuss the details of the Remez algorithm [42] as it is outside of our scope and use an implementation of it to obtain the approximation for the ReLU function. As ReLU is not a differentiable function, we take the SoftPlus function as the continuous approximation of it and use the Remez algorithm to find the polynomial approximation defined in 3.5.

$$ReLU(x) = 0.798083925 + 0.501041262x + 0.072165x^2 - 4.16504756e - 05x^3 \tag{3.5}$$

In Figure 3.7, we can see that for the range of [-5,5] it works as expected.

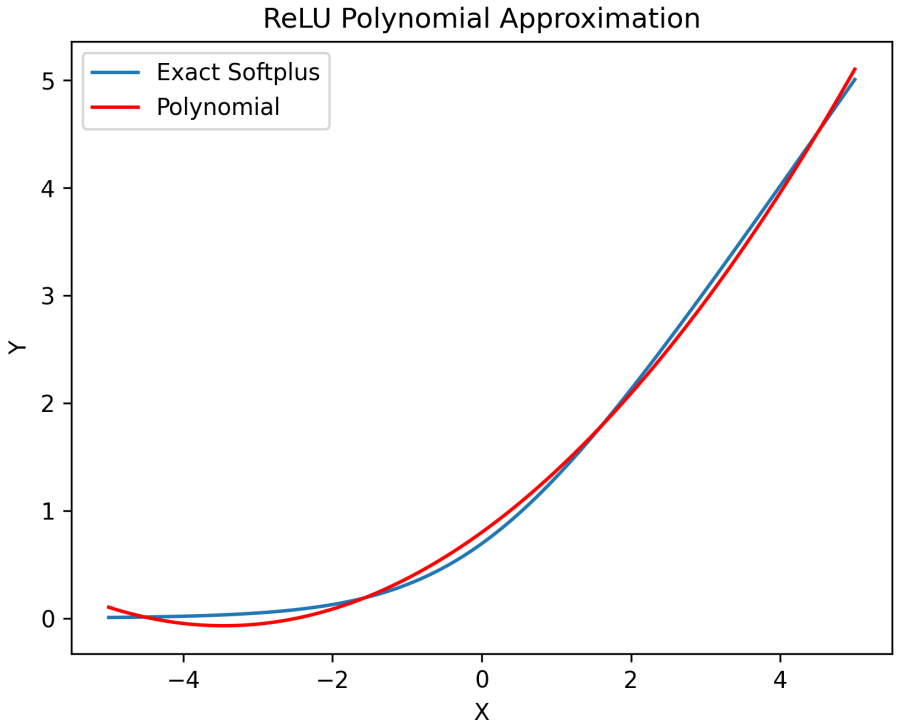

Figure 3.7: SoftPlus polynomial approximation for ReLU approximation

This means that we can use any neural network architecture that uses either ReLU or Sigmoid activation functions and its layers' outputs do not exceed the range [-5,5]. We can see the error distribution in Figure 3.8. The mean error is approximately 0.1, which is pretty high considering that our classification CNN should output a probability between 0 and 1.

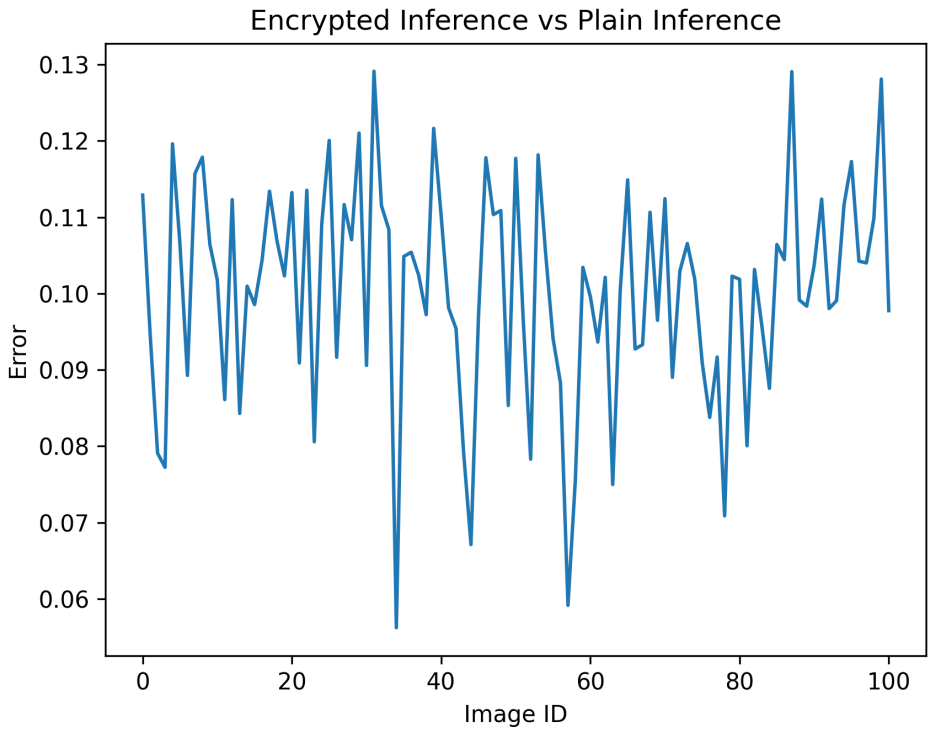

Figure 3.8: Error for encrypted inference versus the plain inference of CNN architecture. The x-axis represents the image IDs of the used testing images. The y-axis shows the error.

However, the error can be manipulated by changing the parameters of the TenSEAL context. Another problem is that the current configurations enable us to use convolution on images no bigger than 20x20 with 3 RGB channels with convolutional kernel sizes 5 and 3, with stride 1. To process bigger images, we may need a bigger than 32768 polynomial modulus degree parameter for the TenSEAL context. This will be inefficient and will have a significant latency issue; furthermore, the maximum size of the poly_mod_degree parameter in the TenSEAL context is 32768. Also, we can make the stride bigger for the convolutional layer; however, that will also cause data loss. Those limitations caused the model architecture on plain images to fail and not learn the patterns well, as by cropping the image to 20x20, we lose a lot of data. If we make the architecture deeper, the trade-off will be the error of encrypted inference as the number of multiplications will increase, as we describe in the sub-

section 3.3.1. That is why we shift from experimenting with CNN architecture to Vision transformers.

3.3.3 Limitations of homomorphic encryption and approximation methods

The encrypted inference architecture is essentially based on the outsourced computing design, whereby the user needs some computation to be performed on the data but doesn't want the data to be shown in plaintext to the third party.

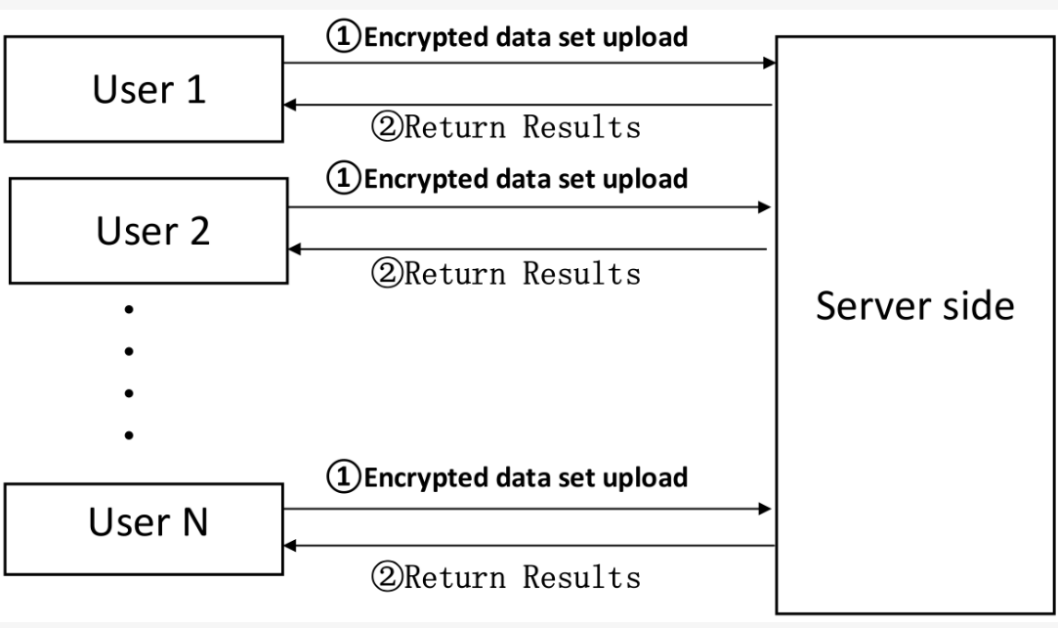

Figure 3.9: Privacy-preserving model for outsourced computing based on homomorphic encryption by Wang et al. [56]

Figure 3.10 describes the client-server side interaction based on the homomorphic encryption model. Due to the fact that encryption is performed by the user on the client-side, the input to the model is in the encrypted form right from the beginning. This fact imposes limitations on the types of computations that can be performed on the input data. In addition, as the intuition behind the HE scheme definition implies, the model selection, available layer, and activation function selection are limited as FHE can only support a subset of operations. Non-linear functions such

as log or sin cannot be used directly, and conversions from non-linear to polynomials are needed. For example, in the case of the common CNN architecture, the convolution operation cannot be directly applied to the encrypted data as it requires kernel passes on the subset of data, which is not accessible due to encryption. As mentioned in Section 3.2, one of the reasons for the choice of the architecture is this inherent disability of the encryption scheme. The advantage of the vision transformer architecture over the plain CNN is that it does not require rearrangements of input data, which are impossible given the current homomorphic encryption algorithms. While the vision transformer architecture overcomes most of the limitations, there are still components in the architecture which require separate treatment. To be more specific, issues arise in two places: the layer normalization and Multi-headed attention layers.

The following passage outlines our proposed solution for approximating the layer normalization operation. In Equation 3.1, the division by an encrypted constant and the square root are two operations that are not supported by the TenSEAL encryption library. A possible solution, as proposed by Chen et al. [8], is to approximate these operations using a neural network. However, in practice, this approach is currently infeasible due to a significant loss of precision. This is primarily because the input vectors passed through the network have a wide range, resulting in imprecise encrypted computations during propagation.

To address this issue, we have attempted normalization techniques to reduce the range of data passing through the network. However, in practice, this approach resulted in the inflation of the neural network weights, leading to imprecise encrypted computations. Our proposed solution instead utilizes the Remez algorithm to approximate the function $f(x) = \frac{1}{\sqrt{x}}$. This approach enables us to approximate the layer normalization operation with high precision while minimizing the impact on the network's performance.

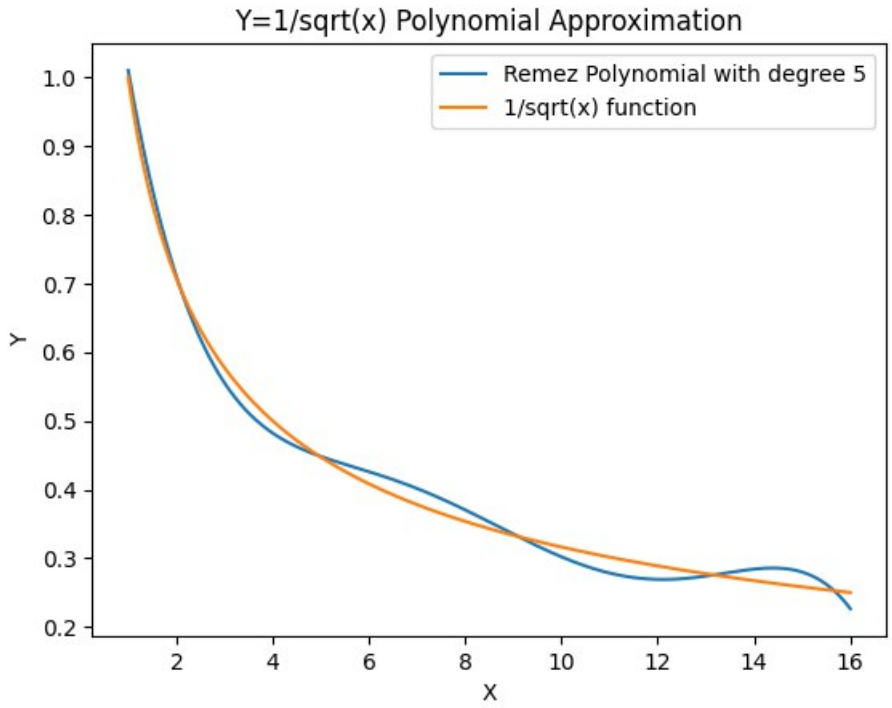

Figure 3.10: Polynomial approximation of the $\frac{1}{\sqrt{x}}$ function using the Remez algorithm.

Figure 3.10 presents the visual representation of the approximating polynomial obtained through the Remez algorithm. We opted to use a degree-5 approximating polynomial since this was the highest degree we could implement with the SEAL encryption library, given the memory limitations of our devices. However, in practice, higher degree polynomials can provide more precise approximations. It is important to note that there is a trade-off between the errors introduced by the multiplicative depth of the encryption and the errors introduced by polynomial approximations. As such, selecting an appropriate degree for the approximating polynomial is critical to balance the precision of the computation with the computational overhead of the encryption scheme.

Table 3.1: Performance of the encrypted Remez approximation for $\frac{1}{\sqrt{x}}$ function.

Encrypted Approximation vs $\frac{1}{\sqrt{(x)}}$	Plaintext Vs Encrypted Approximation
0.01497 ± 0.0072	$4.804e-05 \pm 0.0001$

Table 3.1 illustrates that the average approximation error of the FHE implementation of the proposed polynomial is small, with an average value of 0.01, which is acceptable for our intended task. Moreover, the encrypted difference between the FHE implementation and the plaintext approximation is negligible, further confirming the high precision of our approach.

Given the successful approximation of the $\frac{1}{\sqrt{x}}$ function using the Remez algorithm, it is possible to implement the FHE version of the layer normalization operation by computing the mean and variance using dot product operations on the encrypted vector and using the Remez approximation to compute the denominator in Equation 3.1. This allows for the efficient computation of the layer normalization operation in a privacy-preserving manner.

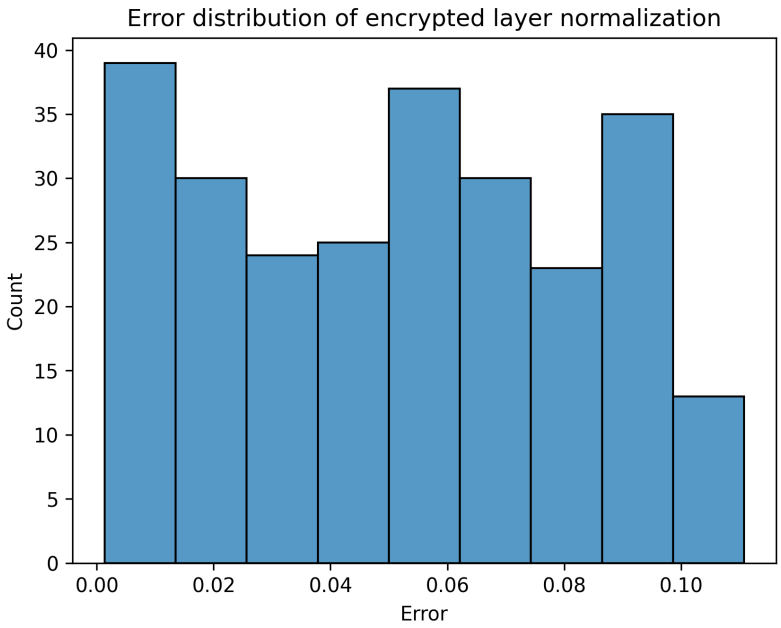

Figure 3.11: Error distribution of the FHE approximation of the layer normalization operation.

Figure 3.11 displays the distribution of absolute error between the layer normalization operation and our encrypted approximation on a randomly generated dataset of floating-point numbers in the range $[0.8, 16]$. This specific range was determined empirically by analyzing the distribution of inputs passing through the layer normal-

ization operation of our vision transformer architecture. The average absolute error of our approximation is 0.05 ± 0.03. The difference between the plaintext and FHE versions of our approximation is 0.0002 ± 0.0001, which is negligible. Therefore, in the subsequent discussions, we used the plaintext implementation of our approximation for the sake of simplicity.

Chapter 4

Results

This chapter presents a comprehensive evaluation of our vision transformer architecture on a retinal scan image dataset. Additionally, we discuss our findings in implementing homomorphic encryption on convolutional neural networks, as well as its inherent limitations. Furthermore, we provide a detailed account of our FHE implementation of the layer normalization operation, which enables efficient privacy-preserving computation of this critical component in deep learning architectures.

Table 4.1: Performance of our proposed model on the RFMiD dataset [1]

Testing dataset	Evaluation Dataset
0.83	0.81

Table 4.1 describes the achieved performance of our vision transformer architecture on the RFMiD dataset. While the resulting accuracy score is significantly lower compared to the RIADD winner architectures, in practice, it is possible to obtain comparable results using ViT architecture as evidenced by Khan et al.[27].

Our second major accomplishment is the successful implementation of encrypted layer normalization, achieved by approximating the root reciprocal function $f(x) = \frac{1}{\sqrt{x}}$ using the Remez algorithm. Our approximation results in a mean absolute error of 0.05 ± 0.03, which is small enough to guarantee that the prediction accuracy of our vision transformer model is not compromised.

Our third achievement is the encrypted implementation of multiple-convolution neural

network (MCNN). Current FHE implementation of MCNN restricts us to processing convolutional layers on images with dimensions of 20x20 and 3 RGB channels and demonstrates a mean error of 0.1.

Chapter 5

Conclusions

In conclusion, this capstone project addressed the issue of privacy in medical imaging, which is a major concern for organizations handling sensitive data. The project focused on usage of machine learning for disease detection on retinal scan images while preserving the privacy of patients. Our findings show that using vision transformers in combination with homomorphic encryption could solve the problem of privacy preservation in medical imaging. The use of vision transformers has demonstrated promising capabilities for image classification tasks, including in our own study where they produced noteworthy results. However, it should be noted that transformer-based models demand a substantial amount of computational resources and intricate implementations. Our research has shown that in order to make them computationally feasible for homomorphic encryption, custom approximation functions may need to be employed to perform complex operations. Future work could explore further use of approximation algorithms to obtain high-acruacy estimations of the activation functions such as Softmax and GeLU to make vision transformers more amenable to homomorphic encryption. Our study provides a basis for further research that will enable secure, high-performance and privacy-preserving machine learning applications in the medical field.

Bibliography

[1] RIADD (isbi-2021) - grand challenge, 2021.

[2] Muhammad Adnan, Sanjay Kalra, James C Cresswell, Shahrad Taheri, Steve Blunsden, Matthew Edey, James Breen, and Andrew Hopper. Federated learning and differential privacy for medical image analysis. *Scientific Reports*, 12(1):1953, 2022.

[3] Ayoub Benaissa, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal. Tenseal: A library for encrypted tensor operations using homomorphic encryption. *CoRR*, abs/2104.03152, 2021.

[4] Alauddin Bhuiyan, Ayyaz Hussain, Ajmal Mian, Tien Yin Wong, Kotagiri Ramamohanarao, and Yasothei Kanagasingam. Biometric authentication system using retinal vessel pattern and geometric hashing. *IET Biom*, 6(2):79–88, 2017.

[5] J. W. Bos, K. Lauter, and M. Naehrig. Private predictive analysis on encrypted medical data. *Journal of Biomedical Informatics*, 50:234–243, Aug 2014.

[6] Jin Chao, Ahmad Al Badawi, Balagopal Unnikrishnan, Jie Lin, Chan Fook Mun, James M. Brown, J. Peter Campbell, Michael F. Chiang, Jayashree Kalpathy-Cramer, Vijay Ramaseshan Chandrasekhar, Pavitra Krishnaswamy, and Khin Mi Mi Aung. CaRENets: Compact and resource-efficient CNN for homomorphic inference on encrypted medical images. *CoRR*, abs/1901.10074, 2019.

[7] Huabo Chen, Ran Gilad-Bachrach, Kyoohyung Han, Zhicong Huang, Arash Jalali, Kimmo Laine, and Kristin Lauter. Logistic regression over encrypted

data from fully homomorphic encryption. *BMC medical genomics*, 11(Suppl 4):81, Oct 2018.

[8] Tianyu Chen, Hangbo Bao, Shaohan Huang, Li Dong, Binxing Jiao, Daxin Jiang, Haoyi Zhou, Jianxin Li, and Furu Wei. The-x: Privacy-preserving transformer inference with homomorphic encryption, 2022.

[9] Jung Hee Cheon, Andrey Kim, Miran Kim, and Yongsoo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017*, pages 409–437, Cham, 2017. Springer International Publishing.

[10] Erfan Darzidehkalani, Mohammad Ghasemi-rad, and P.M.A. van Ooijen. Federated learning in medical imaging: Part i: Toward multicentral health care ecosystems. *Journal of the American College of Radiology*, 19(8):969–974, 2022.

[11] Erfan Darzidehkalani, Mohammad Ghasemi-rad, and P.M.A. van Ooijen. Federated learning in medical imaging: Part ii: Methods, challenges, and considerations. *Journal of the American College of Radiology*, 19(8):975–982, 2022.

[12] A Dehghani, Z Ghassabi, HA Moghddam, et al. Human recognition based on retinal images and using new similarity function. *Journal of Image and Video Processing*, 2013(1):58, 2013.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[15] Judith Sáinz-Pardo Díaz and Álvaro López García. Study of the performance and scalability of federated learning for medical imaging with intermittent clients. *Neurocomputing*, 518:142–154, jan 2023.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[17] Suzuki K. El-Baz A, Gimel'farb G. Machine learning applications in medical image analysis. *Computational and Mathematical Methods in Medicine*, 2017:Article ID 2, 2017.

[18] Hadi Farzin, Hamid Moghaddam, and Mohammad-Shahram Moin. A novel retinal identification system. *EURASIP Journal on Advances in Signal Processing*, 2008, 05 2008.

[19] Danilo Franco, Luca Oneto, Nicolò Navarin, and Davide Anguita. Toward learning trustworthily from data combining privacy, fairness, and explainability: An application to face recognition. *Entropy*, 23(8), 2021.

[20] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[21] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 201–210, New York, New York, USA, 20–22 Jun 2016. PMLR.

[22] M. Goldbaum, S. Moezzi, A. Taylor, S. Chatterjee, J. Boyd, E. Hunter, and R. Jain. Automated diagnosis and image understanding with object extraction, object classification, and inferencing in retinal images. In *Proceedings of 3rd IEEE International Conference on Image Processing*, volume 3, pages 695–698 vol.3, 1996.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[24] Müller Henning, Michoux Nicolas, Bandon David, and Geissbuhler Antoine. A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International Journal of Medical Informatics*, 2004.

[25] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[26] DANIEL HUYNH. Ckks explained, part 3: Encryption and decryption, Sep 2020.

[27] Omar Salman Khan, Ramsha Abbasi, Syed Omer Gilani, and Asim Waris. Ensemble based Multi-Retinal Disease Classification and Application with RFMiD dataset using deep learning. *Biomedical Signal Processing and Control*, 2023. currently under review as far as we are aware.

[28] Hiroaki Kiya, Takuya Nagamori, Satoshi Imaizumi, and Shuhei Shiota. Privacy-preserving semantic segmentation using vision transformer. *J Imaging*, 8(9):233, 2022.

[29] E Sudheer Kumar and C Shoba Bindu. MDCF: Multi-disease classification framework on fundus image using ensemble cnn models. *Journal of Jilin University*, 40(09):35–45, 2021.

[30] Cemal Köse and Cevat İki˙baş. A personal identification system using retinal vasculature in retinal fundus images. *Expert Systems with Applications*, 38(11):13670–13681, 2011.

[31] Kristin Lauter, Michael Naehrig, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *ACM Cloud Computing Security Workshop - CCSW 2011*. ACM, January 2011.

[32] Junghyun Lee, Eunsang Lee, Joon-Woo Lee, Yongjune Kim, Young-Sik Kim, and Jong-Seon No. Precise approximation of convolutional neural networks for homomorphically encrypted data. *ArXiv*, abs/2105.10879, 2021.

[33] Thadaney Israni S. Matheny ME, Whicher D. Artificial intelligence in health care: A report from the national academy of medicine. *JAMA*, 2020.

[34] Kentaro Mihara, Ryohei Yamaguchi, Miguel Mitsuishi, and Yusuke Maruyama. Neural network training with homomorphic encryption. *CoRR*, abs/2012.13552, 2020.

[35] Christian Mouchet, Jean-Philippe Bossuat, Juan Ramón Troncoso-Pastoriza, and Jean-Pierre Hubaux. Lattigo: a multiparty homomorphic encryption library in go. 2020.

[36] Ahmad Zaib Muhammad Imran Razzak, Saeeda Naz. Deep learning for medical image processing: Overview, challenges and the future. In *Classification in BioApps*, pages 323–350. Springer, 2017.

[37] Dominik Müller, Iñaki Soto-Rey, and Frank Kramer. Multi-disease detection in retinal imaging based on ensembling heterogeneous deep learning models, 2021.

[38] George Onoufriou, Paul Mayfield, and Georgios Leontidis. Fully homomorphically encrypted deep learning as a service. *CoRR*, abs/2107.12997, 2021.

[39] Samiksha Pachade, Prasanna Porwal, Dhanshree Thulkar, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, Luca Giancardo, Gwenolé Quellec, and Fabrice Mériaudeau. Retinal fundus multi-disease image dataset (RFMiD): A dataset for multi-disease detection research. *Data*, 6(2), 2021.

[40] Sachin Panchal, Ankita Naik, Manesh Kokare, Samiksha Pachade, Rushikesh Naigaonkar, Prerana Phadnis, and Archana Bhange. Retinal fundus multi-disease image dataset (RFMiD) 2.0: A dataset of frequently and rarely identified diseases. *Data*, 8(2), 2023.

[41] Zheng Qi, AprilPyone MaungMaung, Yuma Kinoshita, and Hitoshi Kiya. Privacy-preserving image classification using vision transformer, 2022.

[42] E I A Remez. General computational methods of chebyshev approximation: The problems with linear real parameters. *Uspekhi Matematicheskikh Nauk*,

17(2(104)):125–190, 1962.

[43] M. A. Rodriguez, H. AlMarzouqi, and P. Liatsis. Multi-label retinal disease classification using transformers, 2022.

[44] Nilanjana Dutta Roy and Arindam Biswas. Fast and robust retinal biometric key generation using deep neural nets. *Multimedia Tools and Applications*, 79(9):6823–6843, 2020.

[45] Microsoft SEAL (release 4.1). `https://github.com/Microsoft/SEAL`, January 2023. Microsoft Research, Redmond, WA.

[46] Lalit M. Sharma, Neeraj; Aggarwal. Automated medical image segmentation techniques. *Journal of Medical Physics*, 2010.

[47] Bertalan Meskó Stan Benjamens, Pranavsingh Dhunnoo. The state of artificial intelligence-based FDA-approved medical devices and algorithms: an online database. *npj Digital Medicine*, 3(1):118, 2020.

[48] K. Suzuki, P. Yan, F. Wang, and D. Shen. Machine learning in medical imaging. *International Journal of Biomedical Imaging*, 2012:123727, 2012.

[49] Kenji Suzuki. Overview of deep learning in medical imaging. *Radiological Physics and Technology*, 2017.

[50] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

[51] Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. On layer normalizations and residual connections in transformers, 2022.

[52] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 09–15 Jun 2019.

[53] Mingxing Tan and Quoc V. Le. Efficientnetv2: Smaller models and faster training, 2021.

[54] Emanuele Trucco, Alfredo Ruggeri, Thomas Karnowski, Luca Giancardo, Edward Chaum, Jean Pierre Hubschman, Bashir al Diri, Carol Y. Cheung, Damon Wong, Michael Abràmoff, Gilbert Lim, Dinesh Kumar, Philippe Burlina, Neil M. Bressler, Herbert F. Jelinek, Fabrice Meriaudeau, Gwénolé Quellec, Tom MacGillivray, and Bal Dhillon. Validating Retinal Fundus Image Analysis Algorithms: Issues and a Proposal. *Investigative Ophthalmology Visual Science*, 54(5):3546–3559, 05 2013.

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[56] Mingyang Wang, Wenbin Zhao, Kangda Cheng, Zhilu Wu, and Jinlong Liu. Homomorphic encryption based privacy preservation scheme for dbscan clustering. *Electronics*, 11(7), 2022.

[57] Wencheng Yang, Song Wang, Hui Cui, Zhaohui Tang, and Yan Li. A review of homomorphic encryption for privacy-preserving biometrics. *Sensors*, 23(7), 2023.