Privacy Preserving Chest X-ray Classification in Latent Space with Homomorphically Encrypted Neural Inference

Jonghun Kim[1,2] ⓘ, Gyeongdeok Jo[1,2], Sinyoung Ra[3], and Hyunjin Park[1,2] ⋆ ⓘ

[1] Department of Electrical and Computer Engineering,
Sungkyunkwan University, Suwon, Korea
[2] Center for Neuroscience Imaging Research,
Institute for Basic Science, Suwon, Korea
[3] Department of Artificial Intelligence, Sungkyunkwan University, Suwon, Korea
{iproj2,oonn1219,nsy0527,hyunjinp}@skku.edu

Abstract. Medical imaging data contain sensitive patient information requiring strong privacy protection. Many analytical setups require data to be sent to a server for inference purposes. Homomorphic encryption (HE) provides a solution by allowing computations to be performed on encrypted data without revealing the original information. However, HE inference is computationally expensive, particularly for large images (e.g., chest X-rays). In this study, we propose an HE inference framework for medical images that uses VQGAN to compress images into latent representations, thereby significantly reducing the computational burden while preserving image quality. We approximate the activation functions with lower-degree polynomials to balance the accuracy and efficiency in compliance with HE requirements. We observed that a downsampling factor of eight for compression achieved an optimal balance between performance and computational cost. We further adapted the squeeze and excitation module, which is known to improve traditional CNNs, to enhance the HE framework. Our method was tested on two chest X-ray datasets for multi-label classification tasks using vanilla CNN backbones. Although HE inference remains relatively slow and introduces minor performance differences compared with unencrypted inference, our approach shows strong potential for practical use in medical images. Our code is available at github.com/jongdory/Latent-HE.

Keywords: Homomorphic Encryption · Chest X-Ray · Classification

1 Introduction

Medical imaging data contain sensitive personal patient information and therefore require strict confidentiality [1]. The analysis and processing of medical images often involve sending data to cloud servers or external computing resources, raising important concerns about data security and privacy [24]. One method

---

⋆ Corresponding Author

(a) Sending images directly to a cloud server
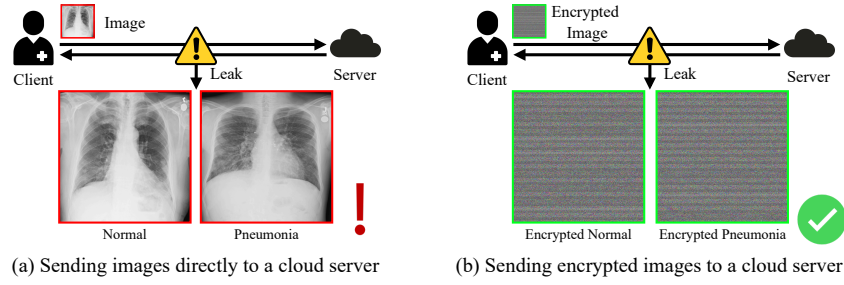(b) Sending encrypted images to a cloud server

Fig. 1. Examples of chest X-ray analysis. (a) Client sends images directly to a server with security risks. (b) Client sends encrypted images to a server without security risks.

to protect privacy when sending data to a server is to encrypt the data before transmission [28,2]. Although this approach eliminates the risk of data leakage, it also requires the encryption and decryption keys to be shared between the server and client. If the secret key of the server is exposed, all the encrypted data can be decrypted [2]. In addition, because the server must decrypt the data to process them, the server can access the information, posing a privacy risk and possibly leading to a reluctance to send data to the server.

One solution is to use homomorphic encryption (HE) [10,4,9,23]. HE allows data to be processed while still encrypted, thereby allowing operations to be performed without decryption. The results remained encrypted and were identical to those obtained from the original decrypted data. This ensures that the data remain confidential during processing. With HE, the server does not need to know the secret key and the client uses only a public key to encrypt the data. The server can then perform the necessary operations on the encrypted data without accessing the actual information, thereby maintaining data privacy. **This also means that even if the data is exposed during processing or transmission, the privacy of the information is still protected.** Fig. 1 compares the risks of sending unencrypted and encrypted data to a cloud server for medical image analysis.

There are two main HE schemes: Boolean and arithmetic. Torus Fully HE (TFHE) is a well-known Boolean operation scheme [6]. With extensions such as programmable bootstrapping, TFHE can perform some basic arithmetic operations; however, it is primarily designed for bit-level tasks. Complex arithmetic involving integers or real numbers requires a cascade of Boolean circuits, which increases computation time. However, Boolean schemes are efficient for simple logic operations (e.g., AND and OR), particularly for short bit lengths or binary logic. Cheon-Kim-Kim-Song (CKKS) [5] is an HE scheme that supports approximate arithmetic. This allows the direct addition and multiplication of encrypted real numbers. Because CKKS approximates the encryption of floating-point numbers, it allows for a small margin of error. Managing a wide range of precision in CKKS requires larger encryption parameters, which significantly slow the calculation [5]. CKKS is commonly used in statistics and
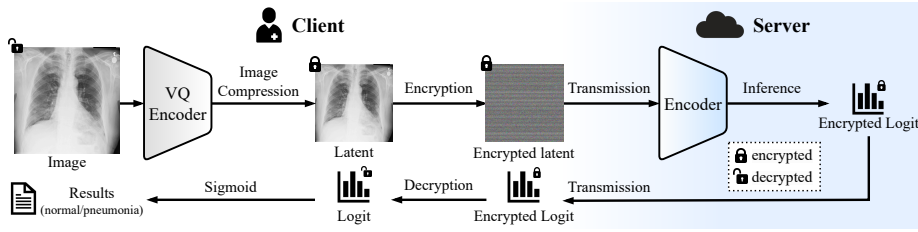
Fig. 2. Illustration of the framework for HE inference.

machine learning [20,12,7]; therefore, it was used in this study. Inference becomes computationally expensive in the HE framework; thus, there is active research on HE inference [22,20,27]. These previous studies only tested HE inference on relatively small images, such as 32×32 images, owing to inference time limitations. In real-world applications, medical images are often much larger, making it difficult to apply these methods directly

To overcome these challenges, this study proposes a framework for medical image inference in HE. First, the images were compressed into smaller latent representations. Subsequently, a classification model was trained on these latent images for analysis. In addition, we adapted efficient computational modules, such as squeeze and excitation blocks [13], which have been effective in improving traditional CNNs for the HE framework. To the best of our knowledge, our study is **one of the first to apply medical image classification of X-ray within the HE framework**, demonstrating that HE can be applied to medical image analysis in which images are large and thus require large parametric models.

2   Method

In this paper, we present a framework for HE inference that can be used in the medical image analysis of X-ray images. Fig. 2 shows the framework for performing HE inferences on a server using an existing classification model. First, only the public key is shared with the server, while the private key remains with the client. On the client side, the image is compressed into smaller latent representations using image compression and then encrypted with a public key before being sent to the server. The server processes the encrypted data without decryption and generates encrypted logits. These encrypted logits are then sent back to the client, where the client decrypts them using a private key and applies a sigmoid function to obtain the final classification results. The background of HE operations is provided in Section 2.1. Image compression is explained in Section 2.2, and the design of the model architecture is described in Section 2.3.

2.1   Background

HE allows computations to be performed on the encrypted data without decryption. After performing operations on the encrypted data and decrypting,
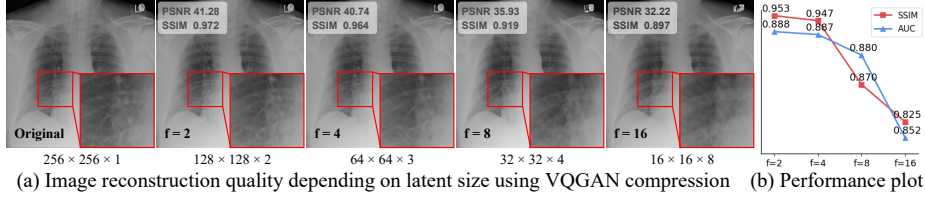
(a) Image reconstruction quality depending on latent size using VQGAN compression   (b) Performance plot

Fig. 3. Image quality, model performance, and HE inference time as a function of latent size. (a) Red bounding boxes indicate areas of interest with enlarged views. (b) Performance plot based on the ResNet 20 model according to the downsampling factor.

the results are the same as if the operations were performed on the original unencrypted data. Specifically, for the encryption and decryption functions $\mathbb{E}$ and $\mathbb{D}$ for two data points x_1 and x_2,

$$x_1 + x_2 = \mathbb{D}(\mathbb{E}(x_1) + \mathbb{E}(x_2)), \quad x_1 \times x_2 = \mathbb{D}(\mathbb{E}(x_1) \times \mathbb{E}(x_2)). \tag{1}$$

Because computations can be performed directly on encrypted data, sensitive information can be processed on a server while remaining encrypted.

2.2   Image Compression

HE inference is computationally intensive, and inference times increase exponentially with the image size. To address this challenge, this study uses a VQGAN [8] to compress images into latent representations. The VQGAN is an effective compression method that generates high-quality latent as shown in recent generative models, such as taming transformers [8] and latent diffusion models [29]. In our approach, images are first compressed into latent using the VQGAN, and classification tasks are then performed on these latent using a classifier. An example of a reconstructed compressed latent is shown in Fig. 3 (a). As expected, there is a trade-off between the size of the latent representation and image quality: higher compression rates lead to lower quality in the reconstructed images. The loss of image quality is also associated with information loss in the latent space, which directly affects classifier performance. As shown in Fig. 3 (b), both the image quality and model performance decreased as the downsampling factor f increased. However, reducing the latent size leads to a more efficient inference. This balance among compression, quality, and performance is crucial for optimizing HE-based inferences in medical image analysis. Given an image x, the latent representation z is obtained as follows:

$$z = \mathbf{q}(E(x)), \quad \hat{x} = G(z), \tag{2}$$

where $\mathbf{q}$ denotes the quantization function, E is the encoder, G represents the decoder (generator), and $\hat{x}$ refers to the reconstructed image. Our VQGAN was trained using the following loss terms.

$$\mathcal{L}_{vqgan} = \underbrace{||x - \hat{x}||^2}_{\text{Reconstruction}} + \underbrace{\log D(x) + \log(1 - D(\hat{x}))}_{\text{GAN}} + \underbrace{\mathcal{L}_{vq}}_{\text{Codebook}}, \tag{3}$$
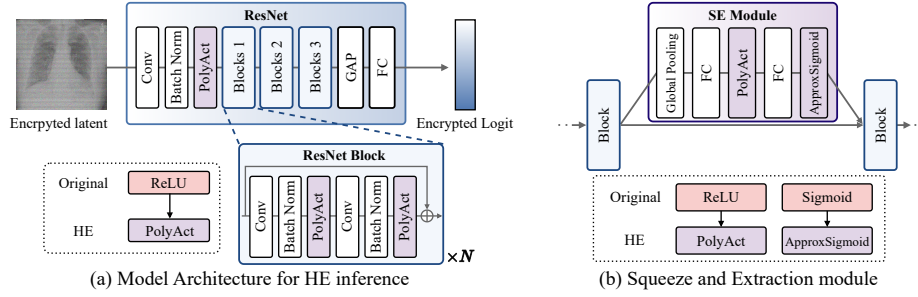
(a) Model Architecture for HE inference   (b) Squeeze and Extraction module

Fig. 4. Changes to activation functions from the existing model for HE inference.

where D denotes the discriminator and $\mathcal{L}_{vq}$ is the quantization loss [32]. In this study, we used latent z as the input to the classifier.

2.3   Architecture Design

HE in the CKKS scheme supports only addition and multiplication and not division or logical operations. Therefore, using activation functions such as sigmoid and ReLU is difficult. One solution is to approximate these activation functions using polynomial functions that only require addition and multiplication [26,27]. We adopted this approach in the present study. Fig. 4 (a) shows the modified structure of ResNet with the modified activation functions for implementation in HE. The activation function implemented using polynomials for the ReLU, polyact, is defined as follows:

$$\text{polyact}(x) = ax^2 + bx + c, \tag{4}$$

where a, b, and c are learnable parameters. Additionally, channel attention modules such as squeeze and excitation (SE) [13], have significantly improved the classification performance of previous CNNs. The SE module efficiently improves the performance without significantly increasing the number of parameters. Therefore, implementing this module in HE with limited computational resources can improve the performance with a small increase in the number of parameters Fig. 4 (b) shows the modified SE module. The existing SE module uses ReLU and sigmoid functions, which we have approximated with polynomial functions. Specifically, the sigmoid function was approximated as follows:

$$\text{approxsigmoid}(x) = \alpha x^3 + \beta x^2 + \gamma x + d, \tag{5}$$

where α, β, γ, and d are all learnable parameters. Higher degrees of accuracy can be achieved when approximating activation functions using higher-degree polynomials. However, in HE, each additional multiplication exponentially increases the computational cost, rendering high-degree polynomials impractical [21]. Therefore, lower-degree polynomials are preferred. However, because a sigmoid module requires a more accurate approximation than ReLU, we use a cubic polynomial for the sigmoid, as in previous studies [12,7].

Table 1. Performance results depending on model architecture for chest X-ray classification. **Micro**, **Macro**, and **Weight** represent the micro, macro, and weighted average, respectively. All models in the table were trained and evaluated at $f = 8$.

Dataset	CheXpert						NIH					
Metric	AUROC ↑			F1 ↑			AUROC ↑			F1 ↑		
Model	Micro	Macro	Weight.	Micro	Macro	Weight.	Micro	Macro	Weight.	Micro	Macro	Weight.
LeNet	0.892	0.835	0.850	0.766	0.606	0.741	0.866	0.693	0.682	0.443	0.047	0.305
LeNet + SE	0.903	0.859	0.870	0.784	0.678	0.774	0.864	0.670	0.688	0.429	0.046	0.300
VGGNet	0.901	0.856	0.868	0.781	0.648	0.764	0.856	0.612	0.649	0.396	0.044	0.283
VGGNet + SE	0.905	0.862	0.875	0.788	0.677	0.779	0.855	0.654	0.658	0.413	0.047	0.293
HefNet	0.909	0.869	0.879	0.796	0.695	0.787	0.856	0.658	0.674	0.425	0.046	0.299
HefNet + SE	0.911	0.870	0.883	0.798	0.694	0.790	0.855	0.690	0.721	0.306	0.104	0.269
ResNet20	0.918	0.880	0.887	0.801	0.696	0.793	0.862	0.690	0.692	0.426	0.110	0.355
ResNet20 + SE	0.920	0.881	0.890	0.803	0.699	0.794	0.869	0.728	0.700	0.437	0.106	0.348
ResNet32	0.919	0.882	0.892	0.804	0.697	0.793	0.869	0.713	0.710	0.417	0.142	0.364
ResNet32 + SE	0.920	0.882	0.893	0.808	0.700	0.795	0.873	0.716	0.705	0.445	0.130	0.363
ResNet44	0.920	0.884	0.893	0.804	0.701	0.794	0.881	0.737	0.732	0.439	0.114	0.361
ResNet44 + SE	0.921	0.885	0.895	0.809	0.704	0.798	0.889	0.759	0.748	0.476	0.112	0.389
ResNet56	0.920	0.882	0.893	0.809	0.700	0.798	0.885	0.734	0.737	0.443	0.080	0.336
ResNet56 + SE	0.924	0.885	0.896	0.812	0.708	0.805	0.893	0.761	0.752	0.486	0.123	0.382

3 Experiments

Implementation Details. For model training, we used PyTorch v2.0.1 [25], and for HE inference, we used TenSEAL [3] based on Microsoft SEAL [30]. The training parameters were set with a batch size of 128 and a learning rate of 2×10^{-4} using the Adam [18] optimizer. We conducted multi-label classification by performing binary classification for each class and compared our models with baseline models including LeNet [19], HefNet [27], VGGNet [31], and ResNet (20, 32, 44, 56) [11]. The model parameters are shown in fig. 5 (c). All models were trained and evaluated with a downsampling factor $f = 8$, considering the computational cost. The codebook dimension $\mathcal{Z}$ of VQGAN was set to 1024.

Datasets. For model training and evaluation, we used CheXpert [14] and NIH [33] chest X-ray datasets, resizing all original images to 256×256 pixels. Both datasets include multilabel classifications for each image. CheXpert, labeled using VisualChexbert [15], consists of 191,027 subjects, 189,116 for training and 1,911 for testing, with 14 classes. The NIH contains 112,120 subjects, of which 111,010 were used for training and 1,110 were used for testing with 15 classes.

4 Results

Impact of model architecture. We evaluated the main architectures used in HE inference and the versions of these models enhanced with SE modules on the CheXpert and NIH datasets. The results are summarized in Table 1. For most models, the addition of the SE module resulted in small performance improvements, demonstrating that the SE module worked effectively, even when approximated for HE. In addition, we observed small performance gains in the

Table 2. Performance results depending on activation functions. All models in the table were trained and evaluated at $f = 8$. σ represents sigmoid function. The results represent inferences from unencrypted setup. ReLU/σ denote the activations used by typical CNNs, and Poly/Appx σ denote the activations approximated by polynomials.

CheXpert	Activation			AUROC ↑			F1 ↑		
Model	ReLU/σ	Poly/Appx σ	Micro	Macro	Weight.	Micro	Macro	Weight.
ResNet20	✓		0.924	0.890	0.898	0.814	0.715	0.802
		✓	0.918	0.880	0.887	0.801	0.696	0.793
ResNet20 + SE	✓		0.926	0.894	0.901	0.816	0.720	0.808
		✓	0.920	0.881	0.890	0.803	0.699	0.794

Table 3. Performance results depending on the inference environment. All models were trained and evaluated at $f = 8$. ✗ indicates inferences from the unencrypted setup.

CheXpert			AUROC ↑			F1 ↑		
Model	HE	Inference Time	Micro	Macro	Weight.	Micro	Macro	Weight.
ResNet20	✗	6.23ms	0.918	0.880	0.887	0.801	0.696	0.793
	✓	76913.72ms	0.912	0.875	0.884	0.796	0.691	0.788
ResNet20 + SE	✗	8.39ms	0.920	0.881	0.890	0.803	0.699	0.794
	✓	92142.15ms	0.915	0.876	0.888	0.799	0.695	0.790

micro average, which aggregates the results across all classes, indicating an overall improvement in the classification performance. The macro average, which assesses how well the model performs in each class, also showed improved results, suggesting that the model handles all classes effectively and uniformly. Furthermore, the weighted metrics, which consider the sample proportions of each class, showed slight increases in both the F1 and AUC scores. These improvements suggest that the SE module not only improves the overall performance but also ensures balanced performance across different classes.

Impact of activation function. For HE inference, we approximated the activation functions using polynomials. However, this approximation can lead to a decrease in the performance. To assess the extent of performance changes, we compared and tested two versions of the activation functions for a representative CNN, ResNet20, in an unencrypted setup (Table 2). When polynomial activation was used for HE inference, the performance decreased slightly; but, the decrease was not significant. The parameters of the activation function were well-trained, allowing it to function effectively and successfully perform X-ray classification.

Impact of HE inference. The CKKS scheme supports real number operations but incurs slight errors in the computational results [5]. Consequently, there may be differences compared with unencrypted operations. We performed inference in both the unencrypted and HE setups, and compared the results for a representative CNN, ResNet20 (Table 3). The changes in performance owing to the inference environment were minor. However, there was a significant difference in the inference time, which increased exponentially with the number of model parameters and latent size in HE. The inference time as a function of the latent
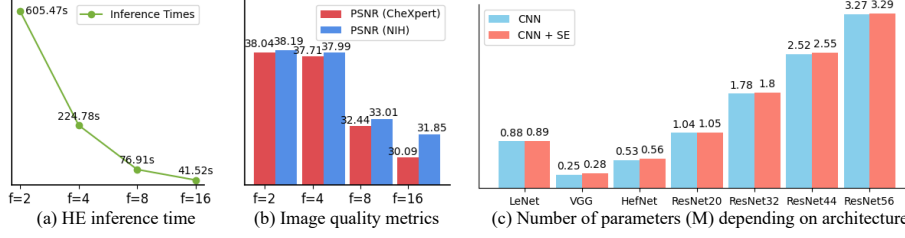
Fig. 5. (a) HE inference time for different latent sizes for ResNet20. (b) Reconstructed image quality in PSNR for different latent sizes. (c) Number of parameters for various model architectures based on $f = 8$.

Table 4. Performance results according to latent size.

CheXpert			AUROC ↑			F1 ↑		
Model	Factor	Latent size	Micro	Macro	Weight.	Micro	Macro	Weight.
ResNet20	f=2	$128 \times 128 \times 2$	0.921	0.888	0.896	0.810	0.719	0.804
	f=4	$64 \times 64 \times 3$	0.920	0.887	0.893	0.810	0.716	0.803
	f=8	$32 \times 32 \times 4$	0.918	0.880	0.887	0.801	0.696	0.793
	f=16	$16 \times 16 \times 8$	0.900	0.852	0.865	0.783	0.665	0.769
ResNet20 + SE	f=2	$128 \times 128 \times 2$	0.923	0.888	0.897	0.813	0.717	0.805
	f=4	$64 \times 64 \times 3$	0.921	0.888	0.896	0.810	0.719	0.804
	f=8	$32 \times 32 \times 4$	0.920	0.881	0.890	0.803	0.699	0.794
	f=16	$16 \times 16 \times 8$	0.911	0.868	0.881	0.791	0.681	0.780

size is shown in Fig. 5 (a). When the downsampling factor was set to $f = 2$, it took almost 10 minwere required to perform the inference using HE on just one sample. For $f = 4$ and $f = 8$, the inference time was significantly reduced.

Impact of latent size. Reducing the latent size makes model training and validation more efficient, but also increases information loss. The changes in the reconstructed image quality based on latent size are shown in Fig. 5 (b). For downsampling factors $f = 2$ and $f = 4$, the image quality was relatively well preserved. However, starting from $f = 8$, the image quality decreased. Similarly, as shown in Fig. 3 (a), images with $f = 4$ retain some lesion details; however, the details are blurred at $f = 8$. To determine whether the loss of information in the latent space leads to a decrease in performance, we validated models with different latent sizes for a representative CNN, ResNet20. The results are summarized in Table 4. At $f = 8$, the reconstructed images lost some detail, but retained sufficient information to prevent a significant performance decrease. However, at $f = 16$, a significant decrease in performance was observed. Therefore, we chose $f = 8$ in this study, considering computational cost and inference time.

5   Discussion

We demonstrate a privacy-preserving framework for HE inference in X-ray image classification. Our study shows that a downsampling factor of eight provides

the best balance between computational cost and image quality. However, the computational cost remains a limitation because HE inference is still relatively slow, and the results are slightly different from those of unencrypted inference. However, the errors introduced were minimal and did not significantly affect the overall performance. In addition, image compression must be performed on the client side, which requires the use of a VQGAN encoder on client devices. Although typical CPU resources can handle compression, processing power from the client is required, which can be limited in certain cases. Despite these challenges, our framework has great potential for practical use in medical imaging while ensuring privacy. Recent research has been conducted on medical image analysis using compressed 3D latent [17,16]. With further improvements in the computational cost, our approach may be effectively applied in real-world settings and 3D medical images.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Andriole, K.P.: Security of electronic medical information and patient privacy: what you need to know. Journal of the American College of Radiology **11**(12), 1212–1216 (2014)
2. Aumasson, J.P.: Serious cryptography: a practical introduction to modern encryption. No Starch Press, Inc (2024)
3. Benaissa, A., Retiat, B., Cebere, B., Belfedhal, A.E.: Tenseal: A library for encrypted tensor operations using homomorphic encryption. arXiv preprint arXiv:2104.03152 (2021)
4. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) **6**(3), 1–36 (2014)
5. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23. pp. 409–437. Springer (2017)
6. Chillotti, I., Gama, N., Georgieva, M., Izabachène, M.: Tfhe: fast fully homomorphic encryption over the torus. Journal of Cryptology **33**(1), 34–91 (2020)
7. Crockett, E.: A low-depth homomorphic circuit for logistic regression model training. Cryptology ePrint Archive (2020)
8. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)

9. Fan, J., Vercauteren, F.: Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive (2012)
10. Gentry, C., Halevi, S.: Implementing gentry's fully-homomorphic encryption scheme. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 129–148. Springer (2011)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hesamifard, E., Takabi, H., Ghasemi, M.: Cryptodl: Deep neural networks over encrypted data. arXiv preprint arXiv:1711.05189 (2017)
13. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
14. Irvin, J., et al.: Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 590–597 (2019)
15. Jain, S., et al.: Visualchexbert: addressing the discrepancy between radiology report labels and image labels. In: Proceedings of the Conference on Health, Inference, and Learning. pp. 105–115 (2021)
16. Kim, J., Na, I., Ko, E.S., Park, H.: Tumor synthesis conditioned on radiomics. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 3635–3646. IEEE (2025)
17. Kim, J., Park, H.: Adaptive latent diffusion model for 3d medical image to image translation: Multi-modal magnetic resonance imaging study. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 7604–7613 (2024)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6980
19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
20. Lee, E., et al.: Low-complexity deep convolutional neural networks on fully homomorphic encryption using multiplexed parallel convolutions. In: International Conference on Machine Learning. pp. 12403–12422. PMLR (2022)
21. Lou, Q., Jiang, L.: She: A fast and accurate deep neural network for encrypted data. Advances in neural information processing systems 32 (2019)
22. Lou, Q., Jiang, L.: Hemet: A homomorphic-encryption-friendly privacy-preserving mobile neural network architecture. In: International conference on machine learning. pp. 7102–7110. PMLR (2021)
23. Marcolla, C., Sucasas, V., Manzano, M., Bassoli, R., Fitzek, F.H., Aaraj, N.: Survey on fully homomorphic encryption, theory, and applications. Proceedings of the IEEE 110(10), 1572–1609 (2022)
24. Mehrtak, M., et al.: Security challenges and solutions using healthcare cloud computing. Journal of medicine and life 14(4), 448 (2021)
25. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems 32 (2019)
26. Peng, H., et al.: Polympcnet: Towards relu-free neural architecture search in two-party computation based private inference. arXiv preprint arXiv:2209.09424 (2022)
27. Ran, R., et al.: Spencnn: orchestrating encoding and sparsity for fast homomorphically encrypted neural network inference. In: International Conference on Machine Learning. pp. 28718–28728. PMLR (2023)

28. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
29. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
30. Microsoft SEAL (release 3.6). `https://github.com/Microsoft/SEAL` (Nov 2020), microsoft Research, Redmond, WA.
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations (ICLR 2015). Computational and Biological Learning Society (2015)
32. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017)
33. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3462–3471 (2017)