

FHE-Assisted Privacy-Preserving Medical Image De-Identification

Jinxiang Dai * and Li S. Cang

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

* Correspondence: jinxiang.dai.ue@163.com

How To Cite: Dai, J.; Cang, L.S. FHE-Assisted Privacy-Preserving Medical Image De-Identification. *Journal of Emerging Technologies with Industrial Applications* **2026**, *1*(2), 9. <https://doi.org/10.53941/jetia.2026.100009>

Received: 17 July 2026

Revised: 17 August 2026

Accepted: 10 October 2026

Published: 21 October 2026

Abstract: Medical image de-identification is essential for data sharing, but conventional OCR-based redaction can expose protected health information (PHI) to the service that performs redaction. We present a client-cloud framework that applies fully homomorphic encryption (FHE) to privacy-preserving medical image de-identification. The client localizes text, segments character patches, encrypts normalized inputs with CKKS, and performs final correction and masking, while the cloud evaluates an FHE-compatible VGG-style character recognizer and CharCNN PHI classifier over ciphertexts. The system reduces encrypted inference cost through polynomial activations, average pooling, diagonal encoding, Toeplitz-style convolution lowering, and BSGS rotation reuse. Prototype results show 100% plaintext-ciphertext prediction consistency, 83.15% corrected character accuracy, and 97.1% encrypted PHI-classification accuracy, with bootstrapped nonlinear layers remaining the main bottleneck.

Keywords: fully homomorphic encryption; CKKS; medical image de-identification; DICOM; optical character recognition; protected health information; privacy-preserving machine learning

1. Introduction

Medical images are increasingly shared across hospitals, public repositories, and cloud platforms for diagnosis, disease screening, treatment assessment, and artificial intelligence (AI) research. This sharing trend reflects a broader movement toward emerging, data-intensive technologies in industrial and organizational analytics, including AI-enabled decision workflows and generative analytics [1,2]. Digital medical images may contain PHI in multiple channels: structured DICOM metadata, private tags, overlays, burned-in annotations, and free text embedded in pixel data. Names, dates, accession identifiers, institution information, scanner information, and patient demographics can therefore remain visible even after simple metadata cleaning. Regulatory frameworks such as HIPAA, GDPR, DICOM confidentiality profiles, and the Personal Information Protection Law of China all motivate robust de-identification before medical image sharing [3–6].

The technical challenge is no longer merely to remove identifiers; it is also to remove them without exposing them to the redaction service. Contemporary image de-identification pipelines often rely on OCR, rule engines, and machine learning classifiers deployed on centralized infrastructure. If the infrastructure is operated by an external cloud vendor or an internal but weakly isolated compute service, the de-identification stage itself becomes a point of privacy exposure. This is especially problematic for large-scale batch processing, cross-institutional studies, and open data preparation workflows, where medical images may be uploaded to a service before PHI is reliably removed.

Fully homomorphic encryption provides a natural cryptographic primitive for this setting. An FHE scheme allows a server to evaluate arithmetic circuits over ciphertexts and return encrypted outputs that decrypt to the same results that would have been obtained in plaintext [7,8]. The CKKS scheme is particularly suitable for approximate real-valued neural network inference because it supports SIMD packing and approximate arithmetic over real or complex vectors [9]. Yet FHE is still expensive for deep learning. Operations that are trivial in plaintext, such as

ReLU, max pooling, comparisons, and argmax, are non-polynomial or discontinuous and require approximation, client-side handling, or alternative network designs.

We develop an FHE-assisted system for medical image de-identification that preserves the practical strengths of OCR-based workflows while reducing plaintext exposure at the cloud. The core design is hybrid. The client performs text localization, character segmentation, pre-processing, decryption, post-processing, and final redaction. The cloud performs compute-intensive recognition and PHI classification over encrypted tensors. This division keeps non-polynomial control operations on the trusted client and reserves the cloud for encrypted neural inference.

The contributions are as follows:

- We define the role of FHE in medical image de-identification by separating confidentiality during outsourced computation from the semantic task of identifying and removing PHI from images and metadata.
- We formulate a client-cloud de-identification architecture in which the cloud performs character recognition and PHI classification without observing plaintext character patches, recovered strings, or PHI-classification features.
- We adapt a VGG11-style character recognizer and CharCNN PHI classifier for CKKS-based encrypted inference by replacing FHE-hostile operators, limiting channel growth, using batch normalization to control activation range, and approximating ReLU through composite minimax polynomials.
- We implement efficient encrypted linear layers and convolutions using diagonal encoding, Toeplitz-style convolution lowering, SIMD packing, and baby-step giant-step (BSGS) rotation reuse, then evaluate the prototype in terms of recognition quality, PHI classification, redaction behavior, communication cost, and runtime.

2. Related Work

2.1. Medical Image De-Identification

DICOM de-identification is governed by both legal requirements and technical standards. The DICOM PS3.15 confidentiality profiles define attribute-level handling patterns for metadata de-identification [4]. The MIDI task group and recent MIDI benchmark work emphasize that safe public release also requires attention to burned-in annotations and pixel-level identifiers, not only metadata [10, 11]. Traditional tools use DICOM tag rules, OCR, heuristics, and manual review [12, 13]. Recent challenge systems have achieved high rule-based scores on benchmark tasks but still operate largely in plaintext [14].

Plaintext OCR-based de-identification is effective when the processing environment is trusted. However, when image batches are outsourced to a cloud service, the cloud may observe exactly the identifiers that the de-identification workflow is meant to remove. This work is complementary to existing de-identification tools: rather than replacing standards-based metadata handling, it addresses the privacy risk of outsourced pixel-text recognition and PHI classification.

2.2. OCR and Character Recognition

OCR systems usually include text detection, orientation classification, and text recognition. PaddleOCR and PP-OCR provide lightweight, open OCR components and have been widely adopted in applied pipelines [15]. In the proposed system, only the detector is retained on the client for text localization. The plaintext recognizer is removed and replaced by an encrypted character recognizer. This design reduces encrypted computation by avoiding FHE-based detection over full-resolution images while ensuring that the cloud receives encrypted character patches instead of raw image regions.

The character classifier is trained on EnglishFnt, a subset of Chars74K containing alphanumeric character images [16, 17]. The prototype extends the character vocabulary with punctuation and “other character” cases to better match DICOM text regions.

2.3. Homomorphic Neural Network Inference

FHE enables encrypted computation but constrains model design. Early neural inference over encrypted data showed that polynomial activations can support private classification but usually requires architecture changes [18]. CKKS improves approximate numerical inference and SIMD packing for real-valued vectors [9]. Practical encrypted CNN inference nevertheless remains dominated by rotations, multiplications, rescaling, and bootstrapping.

Several techniques are central to efficient FHE neural inference. Diagonal methods encode matrix-vector multiplication as slot rotations and componentwise products [19]. BSGS decomposes rotation schedules to reuse intermediate rotations and reduce total rotation count. For nonlinearities, polynomial approximations are used because ReLU and max pooling cannot be evaluated directly as arithmetic circuits. Lee et al. showed that composite minimax polynomial approximations can preserve accuracy for deep CNNs over homomorphic encryption [20]. We apply the same principle to a compact VGG-style character recognizer.

3. Problem Formulation and Threat Model

3.1. Data Model

Let I denote a medical image instance and M its associated DICOM metadata. PHI may appear in M as structured attributes and in I as burned-in text. Let $\mathcal{B} = \{b_i\}_{i=1}^r$ be text bounding boxes localized in I , and let each box be segmented into character patches $\mathcal{X}_i = \{x_{i,j}\}_{j=1}^{n_i}$, where $x_{i,j} \in \mathbb{R}^{32 \times 32}$ after pre-processing. The de-identification objective is to output a redacted image $\tilde{I}$ and sanitized metadata $\tilde{M}$ such that visible or structured PHI is removed while non-PHI diagnostic image content is preserved.

3.2. FHE Protection versus De-Identification

FHE and de-identification address different parts of the privacy problem. FHE protects confidentiality during computation: the cloud evaluates recognition and classification models over ciphertexts and therefore does not receive plaintext character patches, recovered text strings, or plaintext PHI features. De-identification is the content-level operation that determines which visible or structured identifiers should be removed, obscured, or transformed before data sharing. FHE alone does not de-identify an image, and de-identification alone does not prevent a cloud service from seeing PHI during outsourced OCR or classification. The proposed framework combines the two: encrypted inference protects intermediate computation at the cloud, while client-side correction, PHI decisions, metadata handling, and masking produce the de-identified output.

3.3. System Participants

The system has two participants:

- *Client*: trusted by the data owner. The client holds the secret key, performs metadata cleaning, text localization, character segmentation, encryption, decryption, post-processing, and final redaction.
- *Cloud server*: untrusted but computationally capable. The cloud receives public and evaluation keys, encrypted character tensors, and encrypted text-classification tensors. It performs encrypted neural inference and returns encrypted logits or scores.

3.4. Threat Model

The cloud is honest-but-curious: it follows the protocol but attempts to infer PHI from all values it observes. The cloud sees public keys, evaluation keys, bootstrapping keys, ciphertexts, ciphertext sizes, message counts, and coarse timing information. It does not see the secret key or plaintext image regions. Security against ciphertext content recovery relies on the semantic security of the chosen CKKS parameter set under ring-learning-with-errors assumptions.

The protocol protects ciphertext contents but leaves structural leakage through the number of text regions, string lengths, and processing time unless padded batching and constant-schedule processing are added. Malicious-server verifiability is reserved for future work. These limitations are discussed in Section 6.

4. Proposed FHE De-Identification Framework

Figure 1 summarizes the proposed client-cloud workflow. The architecture separates low-cost, non-polynomial, and image-format-sensitive operations from high-cost neural inference. DICOM metadata de-identification and final masking remain local. Encrypted inference is used for the two stages in which cloud acceleration is most useful: character recognition and PHI classification.

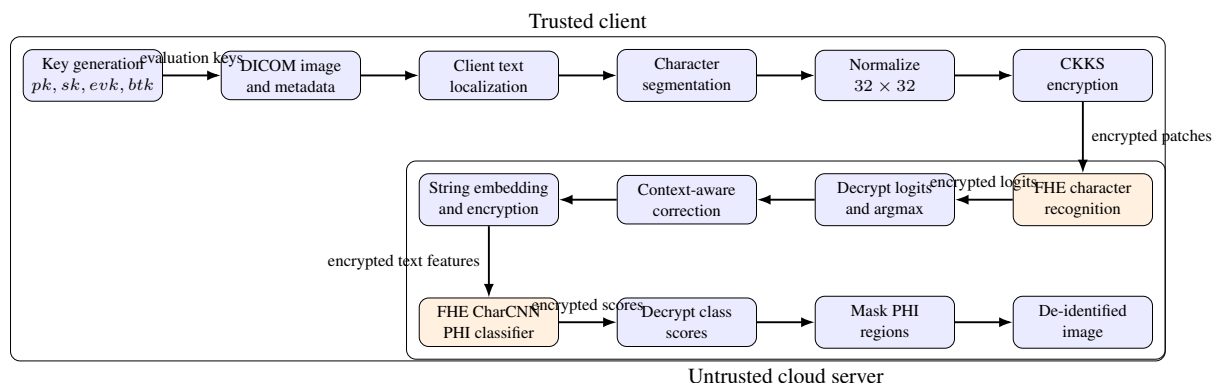


Figure 1. Client-cloud workflow for FHE-assisted medical image de-identification. The cloud evaluates recognition and PHI classification over ciphertexts, while the client keeps secret keys and performs non-polynomial control operations and final redaction.

4.1. Key Generation and Encrypted Upload

The client runs

$$(pk, sk, evk, btk) \leftarrow \text{KeyGen}(1^\lambda), \quad (1)$$

where pk is the public key, sk is the secret key, evk contains evaluation and rotation keys, and btk denotes bootstrapping keys. The cloud receives pk , evk , and btk but not sk . Each normalized character patch x is encoded into CKKS slots and encrypted as

$$ct_x = \text{Enc}_{pk}(\text{Encode}(x)). \quad (2)$$

4.2. Client-Side Text Localization

Full-image FHE text detection would be prohibitively expensive because it requires high-resolution convolutional processing, thresholding, non-maximum suppression, and geometric filtering. Instead, the client uses a trimmed OCR detector. In the prototype, PaddleOCR is pruned to retain the text detection module while removing orientation classification and plaintext text recognition. The detector outputs candidate text boxes that may contain PHI.

4.3. Character Segmentation and Pre-Processing

For each text box, the client applies multi-stage vertical projection segmentation. An initial projection separates character intervals. Intervals whose widths indicate likely connected components are split again. Each character patch is denoised, padded, and resized to 32×32 . This segmentation step converts sequence recognition into a stream of single-character image classification problems, which substantially reduces the encrypted inference burden.

4.4. Encrypted Recognition and Post-Processing

The cloud evaluates an encrypted character classifier and returns encrypted class scores. The client decrypts the scores, applies $\arg \max$, and reconstructs the string. Because single-character recognition errors often follow predictable visual confusions, a lightweight correction stage uses context-specific substitutions. For example, isolated letters in numeric strings may be mapped through a confusion set such as $\{O \leftrightarrow 0, I/l \leftrightarrow 1, Z \leftrightarrow 2, S \leftrightarrow 5, B \leftrightarrow 8\}$ when the surrounding token strongly indicates a numeric field. Similarly, rare digits inside alphabetic DICOM terms can be corrected when the resulting token is a valid medical or DICOM-related word.

4.5. Encrypted PHI Classification and Redaction

After correction, candidate strings are classified as PHI or non-PHI. The client computes the character embedding input for a CharCNN model and encrypts the embedding tensor. The cloud evaluates the CharCNN over ciphertexts and returns encrypted class scores. The client decrypts the scores and masks boxes classified as PHI. This final local redaction prevents the cloud from receiving the original image or the final visible PHI masks.

4.6. FHE-Compatible Neural Inference

4.6.1. CKKS Packed Arithmetic

CKKS encodes a vector of approximate real numbers into a polynomial plaintext and encrypts it into a ciphertext whose slots support SIMD arithmetic. Slotwise additions and multiplications are direct, but cross-slot aggregation requires rotations. If $R_k(x)$ denotes a cyclic rotation by k slots, then an encrypted matrix-vector product can be expressed using diagonal encoding:

$$Ax = \sum_{k=0}^{n-1} d_k \odot R_k(x), \quad (3)$$

where d_k is the k th cyclic diagonal of A , and $\odot$ is componentwise multiplication. Homomorphically,

$$\text{Eval}_A(ct_x) = \sum_{k \in \mathcal{K}} d_k \odot \text{Rot}(ct_x, k), \quad (4)$$

where $\mathcal{K}$ contains the nonzero diagonals.

4.6.2. BSGS Rotation Reuse

Equation (3) may require many rotations. Let $n = m\ell$ and decompose $k = i + jm$. BSGS groups rotations into baby steps i and giant steps jm :

$$Ax = \sum_{j=0}^{\ell-1} R_{jm} \left(\sum_{i=0}^{m-1} d_{i+jm} \odot R_i(x) \right). \quad (5)$$

The inner baby-step rotations are reused across multiple diagonals, reducing the number of distinct rotations from linear in n toward $O(m + \ell)$ under dense diagonal use. This optimization is important for fully connected layers, lowered convolutions, and one-dimensional CharCNN convolutions.

4.6.3. Convolution Lowering

A two-dimensional convolution can be represented as a sparse structured matrix. For an input feature map, each sliding kernel position corresponds to one row of a Toeplitz-like matrix. Kernel weights are placed in columns matching the pixels covered by that receptive field; other entries are zero. Same-padding convolution is handled by adding rows and columns corresponding to padded positions. Once lowered, convolution becomes an encrypted matrix-vector multiplication and can reuse diagonal encoding and BSGS scheduling.

4.6.4. Polynomial Activation

Standard VGG uses ReLU,

$$\text{ReLU}(x) = \max(0, x) = \frac{x(1 + \text{sgn}(x))}{2}. \quad (6)$$

The sign function is discontinuous and cannot be evaluated exactly by CKKS arithmetic. The proposed model approximates it with a composite minimax polynomial:

$$\widehat{\text{sgn}}(x) = p_3(p_2(p_1(x))), \quad (7)$$

where the prototype follows a three-stage degree schedule inspired by precise FHE-CNN approximation work [20]. The encrypted activation is then

$$\widehat{\text{ReLU}}(x) = \frac{x(1 + \widehat{\text{sgn}}(x))}{2}. \quad (8)$$

Composite approximation improves accuracy over a single low-degree polynomial, but it consumes multiplicative depth and frequently requires bootstrapping. The runtime analysis in Section 5 confirms that this activation is the principal computational bottleneck.

4.6.5. FHE-Friendly Character Recognition Network

The character recognizer is derived from VGG11 [21] but modified for encrypted inference. The input is a 32×32 character patch. Channel counts are restricted to at most 64, batch normalization is inserted to stabilize activation ranges, ReLU is replaced by the polynomial approximation above, and max pooling is replaced by average pooling. Average pooling, batch normalization, fully connected layers, and convolution lowering are all linear operations and therefore map naturally to encrypted additions, plaintext-ciphertext multiplications, rotations, and rescaling.

The network architecture and the corresponding encrypted operations at each stage are summarized in Table 1.

Table 1. FHE-Friendly Character Recognition Network.

Stage	Output Shape	Encrypted Operation
Input	$32 \times 32 \times 1$	CKKS packed slots
Conv1 + BN + act.	$32 \times 32 \times 16$	Lowered convolution + poly.
AvgPool	$16 \times 16 \times 16$	Linear averaging
Conv2 + BN + act.	$16 \times 16 \times 32$	Lowered convolution + poly.
AvgPool	$8 \times 8 \times 32$	Linear averaging
Conv3 + BN + act.	$8 \times 8 \times 64$	Lowered convolution + poly.
Conv4 + BN + act.	$4 \times 4 \times 64$	Lowered convolution + poly.
Flatten/FC	1024	Diagonal matvec
FC	1024	Diagonal matvec
Classifier	63 classes	Encrypted logits

4.6.6. Encrypted CharCNN PHI Classifier

The PHI classifier is a one-dimensional character-level CNN inspired by character-level text classification models [22]. The client converts a corrected text candidate into a fixed-length embedding tensor and encrypts it. The cloud evaluates one-dimensional convolutions, pooling-compatible linear reductions, and fully connected layers over ciphertexts. Compared with image convolution, one-dimensional convolution has one fewer spatial dimension and fewer kernel shifts, but it is still implemented through diagonal encoding and rotation reuse.

5. Experimental Evaluation

The evaluation is designed to answer four questions that are central to an encrypted medical image de-identification system. First, does encrypted inference preserve the decisions of the corresponding plaintext model? Second, how much recognition accuracy is lost when OCR is replaced by FHE-compatible character-wise inference? Third, can the encrypted PHI classifier remain useful after text extraction? Finally, where are the computational bottlenecks that limit deployment?

The reported values come from measurements of the implemented prototype. They provide experimental evidence for the proposed design, while full independent reproduction requires disclosure of the CKKS ring dimension, coefficient modulus chain, scale schedule, slot layout, bootstrap parameterization, rotation-key set, random seeds, split construction, and implementation library versions.

The overall evaluation protocol is illustrated in Figure 2.

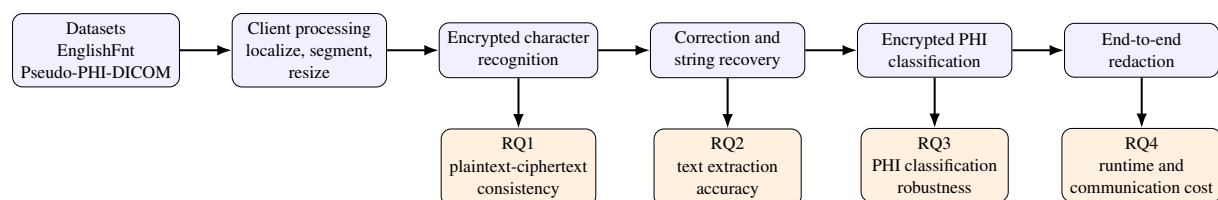


Figure 2. Evaluation protocol. The experiments separate model-equivalence checks, recognition quality, PHI-classification robustness, end-to-end redaction, and system-cost analysis.

5.1. Experimental Setup and Protocol

The client and cloud hardware are summarized in Table 2. The cloud configuration uses high memory and no GPU acceleration, reflecting the difficulty of accelerating the selected FHE operations on commodity GPUs.

Table 2. Experimental Environment.

Side	Component	Configuration
Client	CPU	Intel Core i7-11700 @ 2.50 GHz
	Architecture	x86_64, 64 bit
	Memory	16 GB RAM
	Disk	512 GB
	OS	Windows 11
Cloud	CPU	Intel CPU @ 3.1 GHz
	Architecture	x86
	Memory	128 GB RAM
	Disk	40 GB
	OS	Ubuntu 20.04

EnglishFnt from Chars74K contains 62 alphanumeric classes and 62,992 samples [16,17]. The prototype augments original samples, adds punctuation and other non-alphanumeric examples as an “other” class, and resizes images to 32×32 . For scenario-level testing, samples are selected from Pseudo-PHI-DICOM-Data, which contains 21 subjects, 22 studies, 26 series, and 1,693 DICOM medical images with synthetic PHI-like content. The plaintext-ciphertext consistency evaluation uses 200 validation samples and 200 test samples.

5.2. Metrics

The character recognizer is evaluated using plaintext accuracy Acc_{plain} , encrypted accuracy Acc_{enc} , accuracy drop Acc_{drop} , and prediction consistency (PC):

$$PC = \frac{1}{N} \sum_{i=1}^N \mathbf{1} \left[\hat{y}_i^{plain} = \hat{y}_i^{enc} \right]. \quad (9)$$

End-to-end text extraction is evaluated using character accuracy (CA), string accuracy (SA), and normalized edit distance (NED):

$$CA = \frac{\# \text{correct characters}}{\# \text{ground truth characters}}, \quad (10)$$

$$SA = \frac{\# \text{exactly matched strings}}{\# \text{strings}}, \quad (11)$$

$$NED = \frac{\text{EditDistance}(\hat{s}, s)}{|s|}. \quad (12)$$

For PHI classification and redaction, the evaluation reports accuracy, precision, recall, F1-score, PHI recognition rate, and PHI-F1. Lower NED and lower runtime are better; all other reported metrics are better when higher.

5.3. Plaintext-Ciphertext Consistency

The first experiment verifies that CKKS inference preserves the plaintext classifier's predicted label. This check is important because approximate arithmetic, rescaling, and bootstrapping can perturb logits even when the model architecture is unchanged.

Table 3 and Figure 3 show exact prediction consistency on the reported validation and test splits. The encrypted model matches the plaintext model on every evaluated sample, and the measured accuracy drop is 0.00 percentage points. This result supports using the plaintext model as a faithful proxy during architecture selection and training, provided the encrypted parameterization remains fixed.

Table 3. Plaintext and Encrypted Character Model Consistency.

Split	N	Acc_{plain}	Acc_{enc}	Drop	PC
Validation	200	85.00%	85.00%	0.00%	100.00%
Test	200	87.50%	87.50%	0.00%	100.00%

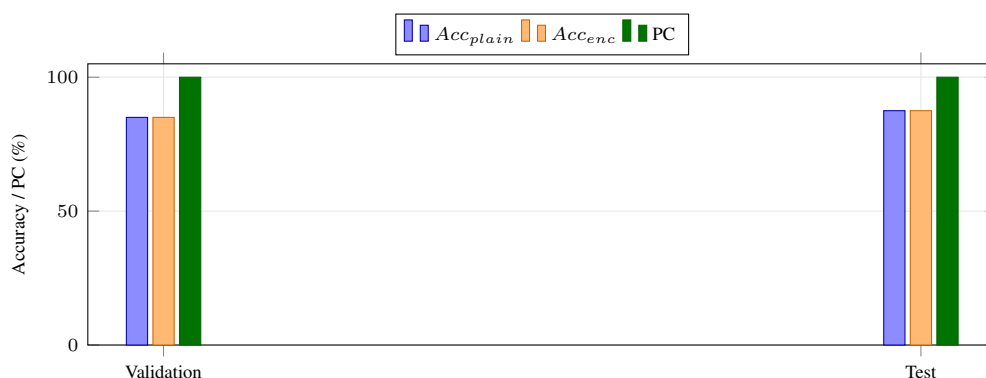


Figure 3. Plaintext and encrypted character-recognition consistency. The identical plaintext and encrypted accuracies indicate that approximate CKKS evaluation does not change the final predicted labels on the reported samples.

5.4. End-to-End Privacy Text Extraction

The second experiment measures the quality of privacy text extraction against PaddleOCR as a plaintext baseline. This comparison is intentionally strict: PaddleOCR uses a mature plaintext OCR recognizer, whereas the proposed method uses segmented character-wise encrypted inference to prevent cloud-side PHI exposure.

Table 4 and Figure 4 show two trends. First, the FHE method has lower raw sequence quality than plaintext OCR, especially at the string level, because segmentation and single-character decisions amplify local errors. Second, domain-aware correction recovers a large fraction of the loss: CA increases from 67.32% to 83.15%, SA

increases from 13.67% to 41.02%, and NED decreases from 0.3942 to 0.2311. The result indicates that deterministic client-side correction is a valuable companion to encrypted character inference.

Table 4. Privacy Text Extraction Performance.

Condition	Method	CA	SA	NED
Before correction	PaddleOCR	98.99%	85.71%	0.0169
	FHE method	67.32%	13.67%	0.3942
After correction	PaddleOCR	98.99%	85.71%	0.0169
	FHE method	83.15%	41.02%	0.2311

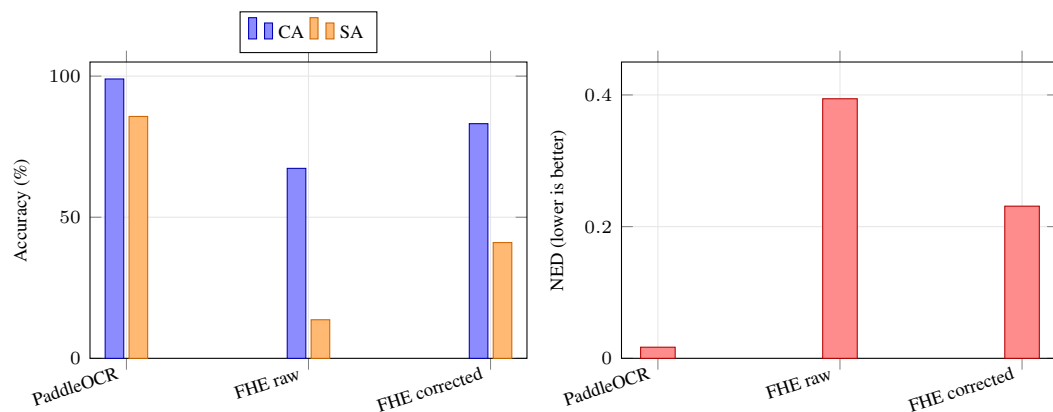


Figure 4. End-to-end privacy text extraction. Context-aware correction substantially improves the encrypted method, reducing normalized edit distance while increasing both character-level and string-level accuracy.

5.5. Runtime, Storage, and Communication Cost

The third experiment profiles the encrypted inference path. Table 5 gives component-level timing and communication cost, while Figure 5 visualizes the bottleneck structure. Six ReLU approximation modules take 90.34 seconds and dominate the neural component profile. At the task level, encrypted inference takes 1622.88 seconds, or 94.1% of the total reported workflow time; communication accounts for 5.8%, and preprocessing accounts for 0.1%.

The storage measurements show why FHE deployment requires careful batching. A 4.10 KB plaintext input expands to a 4882.43 KB ciphertext representation, corresponding to a $1192\times$ expansion and 781,188.81 KB total communication in the reported task. Bootstrapping is also expensive: 12 bootstrap operations require 141.95 seconds in total, with an average time of 11.83 seconds.

Table 5. Runtime and Communication Cost.

Metric	Value	Unit
Convolution layer time	7.60	s
ReLU approximation time	90.34	s
Pooling layer time	2.05	s
Fully connected layer time	0.37	s
Preprocessing time	1.74	s
Communication time	99.46	s
Encrypted inference time	1622.88	s
Plaintext size S_p	4.10	KB
Ciphertext size S_c	4882.43	KB
Expansion ratio E_{enc}	1192.00	$\times$
Total communication C_{total}	781,188.81	KB
Bootstrap operations	12	count
Total bootstrap time	141.95	s
Average bootstrap time	11.83	s

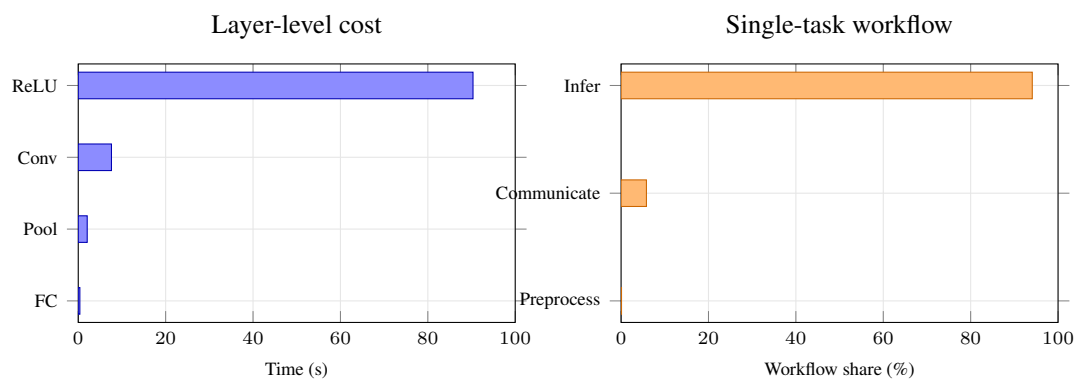


Figure 5. Runtime bottlenecks. Polynomial activation dominates layer-level computation, and encrypted inference dominates the end-to-end workflow.

5.6. PHI Classification and Redaction

The final experiments evaluate whether extracted text can be classified and redacted effectively. The CharCNN PHI classifier is evaluated on SPY-Extended, an extension of a sensitive text dataset that adds common medical personal information fields such as birth date, sex, and visit time. Table 6 shows close agreement between plaintext and encrypted environments. Encrypted accuracy is 97.1%, only 0.2 percentage points lower than plaintext accuracy.

Table 6. CharCNN PHI Classification Performance

Environment	Accuracy	Precision	Recall	F1-Score
Plaintext	0.973	0.965	0.970	0.967
Encrypted	0.971	0.961	0.968	0.964

The complete system is then compared against a comprehensive medical image de-identification tool [12] and Microsoft Presidio [23]. As shown in Table 7 and Figure 6, the FHE method reaches a PHI recognition rate of 75.0%, which is lower than the plaintext baselines by 8.05 and 5.28 percentage points, respectively. Its PHI-F1 score is 74.4%, higher than the comprehensive tool in the reported comparison but lower than Presidio. These results position the method as a privacy-preserving alternative whose main cost is reduced OCR-driven recall rather than failure of the encrypted PHI classifier itself.

Table 7. End-to-End PHI Redaction Comparison

Method	PHI Recognition	PHI-F1
Comprehensive tool	83.1%	71.0%
Presidio	80.3%	82.0%
FHE method	75.0%	74.4%

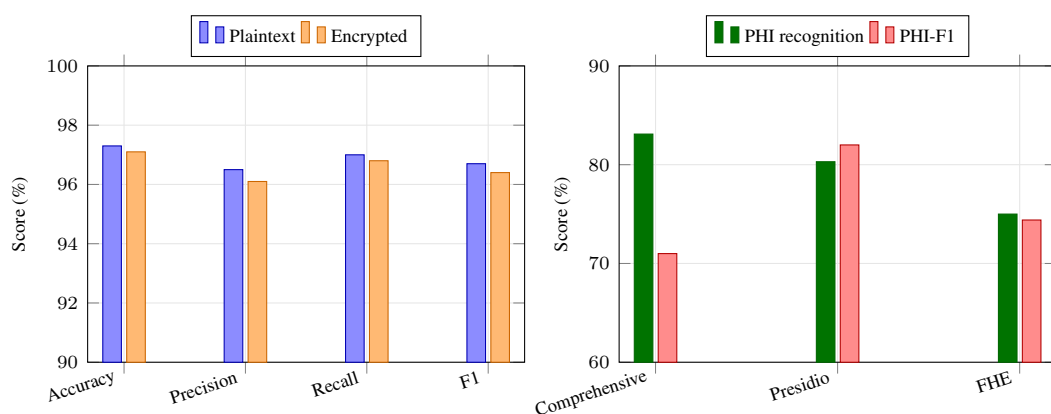


Figure 6. PHI classification and redaction results. The encrypted CharCNN closely matches plaintext classification, while the end-to-end system trades some PHI recognition rate for cloud-side privacy protection.

6. Discussion

6.1. Accuracy-Privacy Tradeoff

The experiments reveal the central tradeoff of encrypted de-identification. Plaintext OCR remains substantially more accurate for text extraction, especially at the string level. The FHE method sacrifices recognition accuracy because it decomposes sequence recognition into segmented single-character classification, uses FHE-compatible approximations, and avoids the richer sequence models used by modern OCR recognizers. However, this loss buys a different security property: the cloud does not observe raw character images, recognized PHI strings, or PHI classification features.

6.2. Why Hybrid Processing Is Necessary

The proposed system is deliberately hybrid. Fully encrypted end-to-end OCR would require encrypted detection, geometric filtering, sequence decoding, comparisons, and argmax. These operations are either expensive or poorly matched to CKKS. By placing detection, segmentation, decryption, correction, and masking on the client, the system keeps encrypted computation focused on dense arithmetic kernels where FHE is most mature.

6.3. Bottlenecks

The measured workflow is dominated by encrypted inference. ReLU approximation and bootstrapping account for most of the component runtime, while ciphertext expansion drives communication cost. These results suggest four optimization directions:

- Replace VGG-style depth with shallower or activation-sparse architectures designed directly for FHE.
- Improve CKKS packing so that more characters or channels are processed per ciphertext.
- Use operator-level parallelism across character streams and across independent rotations.
- Reduce nonlinearity depth through quantization-aware, polynomial-aware, or neural architecture search methods.

6.4. Security Limitations

The cloud sees ciphertext sizes, the number of processed character patches, and timing. In medical images, the existence and length of text regions can reveal weak contextual information even without plaintext. A deployment should pad batches, normalize string lengths, aggregate jobs, and use constant-schedule processing where practical. The current design also assumes an honest-but-curious cloud. Verifiable FHE, secure audit logs, trusted execution environments (TEEs), or multiparty computation (MPC) could be added to address malicious behavior and side-channel resistance.

6.5. Clinical Validation and Real-World Applicability

The prototype has not yet been evaluated in a prospective clinical trial or routine clinical deployment. This is an important scope boundary because medical image de-identification must remain reliable across scanner vendors, modalities, acquisition protocols, annotation styles, languages, and local workflow conventions. Real deployments would also need to quantify how missed PHI, over-redaction, and text-localization failures affect downstream clinical review, research reuse, and audit procedures.

A clinical validation study should therefore include diverse images from multiple departments, representative burned-in annotation patterns, and independent review by clinical and information-governance staff. The evaluation should measure PHI recall, false redaction of diagnostically relevant content, turnaround time, user correction burden, and compatibility with existing DICOM de-identification workflows. These studies are beyond the present prototype, but they are necessary before the system can be used as a clinical de-identification service.

6.6. Reproducibility Requirements for Submission

For a full submission, the implementation should report the exact CKKS parameter set, security level estimate, scale and modulus chain, bootstrap configuration, packing layout, rotation key set, model hyperparameters, training protocol, public availability of datasets or split manifests, and confidence intervals across repeated trials. These details are essential for independent reproduction and for assessing whether the encrypted deployment satisfies the intended security level.

7. Conclusions

We presented an FHE-assisted medical image de-identification system that combines client-side text localization and redaction with cloud-side encrypted character recognition and encrypted PHI classification. The system uses FHE to protect outsourced computation and uses client-side decisions and masking to produce the de-identified image. The CKKS-compatible VGG-style recognizer and CharCNN classifier use polynomial activations, average pooling, diagonal encoding, convolution lowering, and BSGS rotation reuse. Prototype results show exact plaintext-ciphertext prediction consistency for character classification, high encrypted PHI-classification accuracy, and an end-to-end de-identification workflow that preserves PHI confidentiality from the cloud at the cost of lower OCR accuracy and high runtime. Future work should focus on lower-depth encrypted models, stronger side-channel defenses, improved packing and batching, complete reproducibility reporting, and clinical validation across realistic imaging workflows.

Author Contributions

J.D.: conceptualization, methodology, software, data curation, validation, writing—original draft preparation; L.S.C.: visualization, investigation, supervision, software, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Institutional Review Board Statement

Not applicable.

Informed Consent Statement

Not applicable.

Data Availability Statement

The datasets used in this study (EnglishFnt/Chars74K, Pseudo-PHI-DICOM-Data, and SPY-Extended) are publicly available as cited. Prototype code and configuration will be made available upon reasonable request.

Conflicts of Interest

The authors declare no conflict of interest.

Use of AI and AI-Assisted Technologies

During the preparation of this work, the authors used Grammarly AI to check grammar and improve sentence clarity in this manuscript. After using it, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

1. Qin, L. Shaping the Future: Emerging Technologies and Their Role in Industry 4.0 and Beyond. *J. Emerg. Technol. Ind. Appl.* **2026**, *1*, 4. <https://doi.org/10.53941/jetia.2026.100004>.
2. Li, J.; Fu, L.; Wang, Y. Generative AI and management analytics: Survey. *J. Manag. Anal.* **2026**, *13*, 217–263. <https://doi.org/10.1080/23270012.2026.2652312>.
3. U.S. Department of Health and Human Services. Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the HIPAA Privacy Rule. Available online: <https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/> (accessed on 29 June 2026).
4. NEMA. Digital Imaging and Communications in Medicine (DICOM), Part 15: Security and System Management Profiles. Available online: <https://dicom.nema.org/medical/dicom/current/output/html/part15.html> (accessed on 29 June 2026).
5. European Parliament and Council of the European Union. Regulation (EU) 2016/679: General Data Protection Regulation. Available online: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> (accessed on 29 June 2026).
6. Standing Committee of the National People's Congress. Personal Information Protection Law of the People's Republic of China. Available online: https://en.npc.gov.cn.cdurl.cn/2021-12/29/c_694559.htm (accessed on 29 June 2026).
7. Rivest, R.L.; Adleman, L.; Dertouzos, M.L. On Data Banks and Privacy Homomorphisms. In *Foundations of Secure Computation*; Academic Press: New York, NY, USA, 1978; pp. 169–180.

8. Gentry, C. Fully homomorphic encryption using ideal lattices. In Proceedings of the STOC '09: Symposium on Theory of Computing, Bethesda, MD, USA, 31 May–2 June 2009; pp. 169–178. <https://doi.org/10.1145/1536414.1536440>.
9. Cheon, J.H.; Kim, A.; Kim, M.; et al. Homomorphic Encryption for Arithmetic of Approximate Numbers. In *Advances in Cryptology – ASIACRYPT 2017*; Springer International Publishing: Cham, Switzerland, 2017; Volume 10624, pp. 409–437. https://doi.org/10.1007/978-3-319-70694-8_15.
10. Clunie, D.A.; Flanders, A.; Taylor, A.; et al. Report of the Medical Image De-Identification (MIDI) Task Group—Best Practices and Recommendations. *arXiv* **2023**, arXiv:2303.10473.
11. Pei, L.; Sutton, G.; Rutherford, M.; et al. Medical Image De-Identification Benchmark Challenge. *arXiv* **2025**, arXiv:2507.23608v1.
12. Rempe, M.; Heine, L.; Seibold, C.; et al. De-Identification of Medical Imaging Data: A Comprehensive Tool for Ensuring Patient Privacy. *arXiv* **2024**, arXiv:2410.12402.
13. Naddeo, K.; Koutsoubis, N.; Krish, R.; et al. DICOM De-Identification via Hybrid AI and Rule-Based Framework for Scalable, Uncertainty-Aware Redaction. *arXiv* **2025**, arXiv:2507.23736v1.
14. Jiang, H.; Xie, S.; Wan, Z. A DICOM Image De-identification Algorithm in the MIDI-B Challenge. *arXiv* **2025**, arXiv:2508.07538.
15. Du, Y.; Li, C.; Guo, R.; et al. PP-OCR: A Practical Ultra Lightweight OCR System. *arXiv* **2020**, arXiv:2009.09941.
16. de Campos, T.E.; Babu, B.R.; Varma, M. Character Recognition in Natural Images. In Proceedings of the International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, 5–8 February 2009; pp. 273–280. <https://doi.org/10.5220/0001770102730280>.
17. Centre for Vision, Speech and Signal Processing, University of Surrey. The Chars74K Dataset. Available online: <https://info-ee.surrey.ac.uk/CVSSP/demos/chars74k/> (accessed on 29 June 2026).
18. Gilad-Bachrach, R.; Dowlin, N.; Laine, K.; et al. Crypto-Nets: Neural Networks over Encrypted Data. *arXiv* **2014**, arXiv:1412.6181v2.
19. Halevi, S.; Shoup, V. Algorithms in HELib. In *Advances in Cryptology—CRYPTO 2014*; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8616, pp. 554–571. https://doi.org/10.1007/978-3-662-44371-2_31.
20. Lee, J.; Lee, E.; Lee, J.-W.; et al. Precise Approximation of Convolutional Neural Networks for Homomorphically Encrypted Data. *arXiv* **2021**, arXiv:2105.10879v4.
21. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556v6.
22. Zhang, X.; Zhao, J.; LeCun, Y. Character-level Convolutional Networks for Text Classification. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 7–12 December 2015; Volume 28, pp. 649–657.
23. Data Privacy Stack. Presidio: Data Protection and De-Identification SDK. Available online: <https://github.com/data-privacy-stack/presidio> (accessed on 29 June 2026).