Jia *et al. Journal of Cloud Computing* (2023) 12:175
https://doi.org/10.1186/s13677-023-00537-0

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

Efficient and privacy-preserving image classification using homomorphic encryption and chunk-based convolutional neural network

Huixue Jia[1], Daomeng Cai[2,3], Jie Yang[3], Weidong Qian[4], Cong Wang[5*], Xiaoyu Li[1] and Shan Yang[6]

Abstract

Image feature categorization has emerged as a crucial component in many domains, including computer vision, machine learning, and biometrics, in the dynamic environment of big data and cloud computing. It is extremely difficult to guarantee image data security, privacy, and computing efficiency while also lowering storage and transmission costs. This paper introduces a novel method for classifying image features that combines multilevel homomorphic encryption and image data partitioning in an integrated manner. We employ a novel partitioning strategy to reduce computational complexity, significantly reducing computational load and improving classification accuracy. In the quest for increased data security and privacy, we introduce a novel, fully homomorphic encryption approach specialized to partitioned images. To counter the inherent complexity of encryption, we devise a compound encryption strategy that exploits the full potential of homomorphic computation, with an explicit objective to curtail computational and storage overheads. Evidently superior to conventional methods, our methodology showcases pronounced benefits in computational efficiency, storage and transmission cost reduction, and robust security and privacy preservation. Hence, the methodology put forth in this paper presents a pioneering and efficacious resolution to the multifaceted challenges of image feature classification within the intricate milieu of cloud computing and big data.

Keywords Homomorphic encryption, Image chunking, Image classification, Convolutional neural networks

Introduction

Deep learning has become integral to image classification applications, including facial recognition, object detection, and information forensics. These applications demand substantial computational resources and large-scale data sets to ensure optimal performance (Fig. 1). Centralized training, wherein individual training samples are uploaded to a third-party entity with extensive computational resources (e.g., cloud servers) to develop the final model, is the conventional approach, as shown in Fig. 2a. However, this method raises significant privacy concerns, as these third-parties might misuse sensitive data from multiple users.

To address these concerns, distributing training tasks to the users is a plausible solution. This approach enables users to train the model locally and share only the gradients, thus eliminating the need to disclose private data, as shown in Fig. 2b. While this reduces privacy risks, attackers can still infer the attributes of the samples from the shared gradients. This threat is especially pronounced in decentralized learning systems such as the one shown in Fig. 2c, as any user connected to honest users can access

*Correspondence:
Cong Wang
cong-wang@foxmail.com
[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China
[2] School of Mechanical Engineering and Automation Beihang University, Beijing, China
[3] CSSC System Engineering Research Institute, Beijing, China
[4] China Ship Scientific Research Center, Wuxi, China
[5] The Intelligent Policing Key Laboratory of Sichuan Province, Sichuan Police College, Luzhou, China
[6] Department of Chemistry, Physics, and Atmospheric Sciences, Jackson State University, Jackson, Mississippi, USA

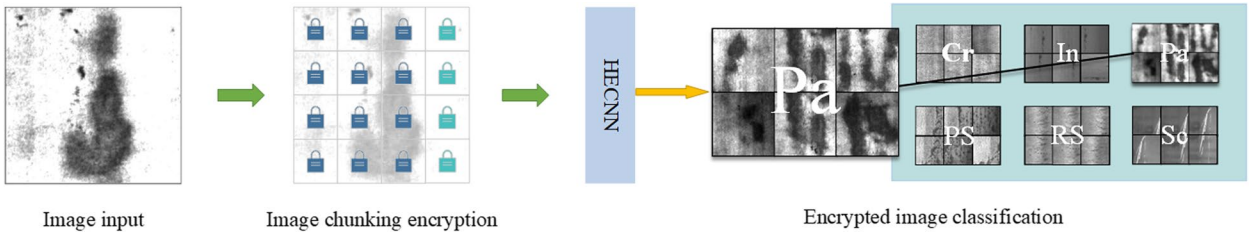

Fig. 1 It shows how a modified convolutional neural network is used to process the original images after they have been encrypted using the Completely Homomorphic Encryption (CKKS) technique

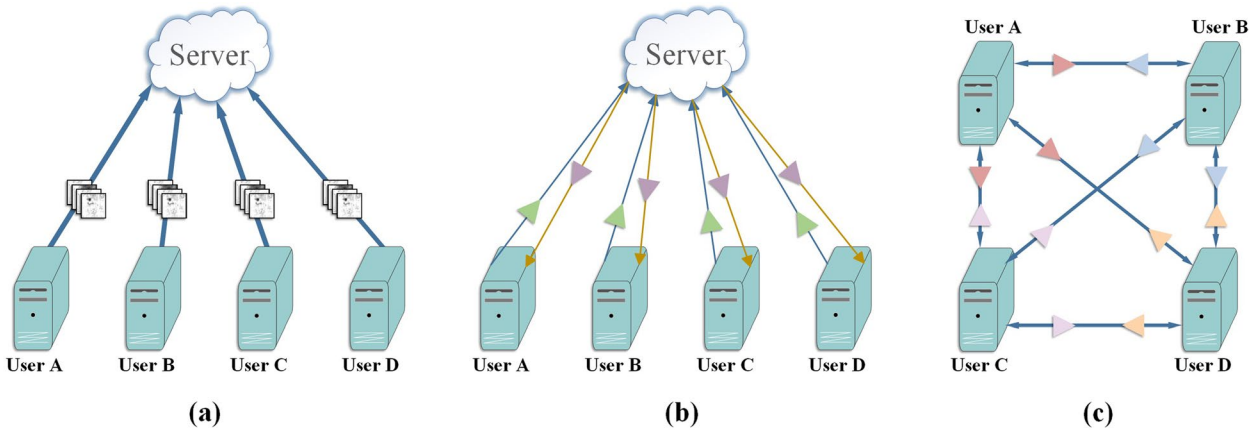

Fig. 2 Illustration of various deep learning models. **a** Centralized Training: Depicts a model where users upload their individual training samples to a third-party entity, typically a cloud server with substantial computational resources, which is responsible for generating the final model. **b** Federated Learning Systems: Demonstrates an approach where models are trained locally, with only the gradients being shared, thereby ensuring that private data remains unexposed. **c** Decentralized Learning System: Represents a system where users have the autonomy to exchange gradients with their neighboring users and independently update their models

their gradients and potentially compromise their data privacy [1].

In the era of cloud computing and big data, the outsourcing of data to cloud servers is becoming increasingly prevalent, thereby reducing the consumption of local storage and computational resources [2]. However, with the proliferation of malicious users on the internet and the inherent unreliability of cloud service providers, privacy and security concerns are paramount. Consequently, the development of an efficient and privacy-centric deep learning framework is of utmost importance [3].

Homomorphic encryption, which allows computation on encrypted data, is a promising solution to this issue. However, its practical application in deep learning is limited due to its high computational complexity. This paper presents a novel privacy-protecting image feature extraction method that integrates homomorphic encryption with deep learning techniques. By leveraging the inherent sparsity of image data, we reduce the computational burden of fully homomorphic encryption while maintaining the privacy of sensitive image data.

Homomorphic encryption is utilized to protect the privacy of user data operated on in cloud computing. Current research does not explicitly outline the specific method of segmenting images, computing gradient values, applying homomorphic encryption only to blocks containing important features, and then classifying the encrypted blocks. Therefore, the method in this paper is an optimized solution for image classification problems.

In our study, we confirm the effectiveness of our proposed approach through an empirical case study involving the utilization of a modified convolutional neural network (CNN) model [4]. This model, known for its capabilities in image recognition tasks, is the backbone of our approach. We pre-process the input image. This critical stage ensures that the image is in the best format for subsequent processing. After pre-processing, we divide the image into different chunks. This allows us to apply our method to each chunk independently, thus reducing computational complexity and increasing scalability. Once the image is partitioned into chunks, we apply a fully homomorphic encryption scheme to the chunks whose gradients exceed a certain threshold [5].

This selective encryption strategy ensures that we focus our computational resources on the most information-rich parts of the image, thus improving the efficiency of our method without compromising the accuracy of the results.

Our work makes several significant contributions (Fig. 3).

- We introduce a novel image splitting encryption method that reduces the time spent on homomorphic encryption computations.
- We propose a composite encrypted image scheme to lower the computational complexity associated with fully homomorphic encryption, while also reducing computational and storage costs.
- We validate our approach using a custom image classification model and assess its performance in terms of accuracy and computational efficiency.

Related work

Homomorphic encryption

Image privacy protection is an important issue in the fields of mobile computing and cloud services. With the proliferation of mobile devices and cloud services, people are increasingly relying on these platforms to store and process personal data, including images. However, this also brings the risk of privacy leakage, as users' data might be accessed or abused by unauthorized third parties. In recent years, the intersection of Homomorphic Encryption (HE) and image classification has garnered widespread attention, primarily due to the growing demand for data privacy in deep learning applications. Some contemporary research has made significant contributions in this area [6].

In a mobile cloud computing environment, homomorphic encryption can serve as a service to protect the privacy of outsourced images. For instance, one study introduced the concept of "Homomorphic Encryption as a Service" for protecting the privacy of outsourced images in mobile cloud computing environments [7]. This setup allows rich mobile applications to run on various mobile devices, while all data processing and storage are conducted in cloud services outside the mobile devices [8].

Homomorphic encryption technology can process ciphertext data while preserving privacy, and can directly search, compute, and statistic ciphertext on the cloud. This technology has four main aspects of application in cloud computing: (1) Retrieving encrypted data in cloud computing [9]. This shows that by using homomorphic encryption, users can protect their image data, while still being able to conduct some basic processing and analysis on this data without decryption.

Cloud computing is widely adopted due to its low cost, high reliability, and generic services. However, as transactions between users and service providers are often asynchronous, data privacy could lead to a crisis of trust, thus hindering the expansion of cloud computing applications. In a paper, the authors proposed a data privacy protection scheme based on homomorphic encryption to address this issue [10].

Overall, homomorphic encryption provides a potent solution for image privacy protection in mobile computing and cloud services. By using homomorphic encryption, the security and privacy of user data can be ensured, while still being able to conduct necessary computations and analyses on the encrypted data.

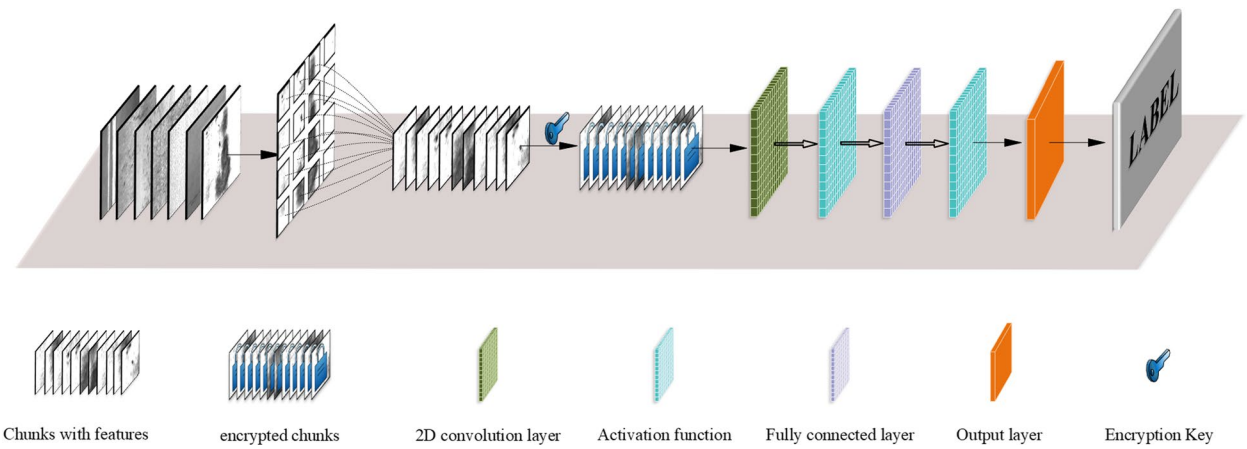

Fig. 3 Depiction of an advanced image processing pipeline. The process initiates with the partitioning of the original image, followed by the full homomorphic encryption of blocks with distinctive features. These encrypted image blocks are then processed by a modified convolutional neural network

Image privacy protection

The issue of privacy protection in image data, particularly facial images, has been a focal point of recent research [11]. Several innovative methodologies have been proposed to address this concern [12], leveraging advanced techniques such as federated learning, ensemble models, and generative adversarial networks (GANs).

Yang et al. [13] proposed a transferable face image privacy protection method based on federated learning and ensemble models. Their approach utilizes a federated learning model established on distributed datasets and a local facial recognition model obtained through local face data training. The method demonstrated its effectiveness in generating private face images with high transferability and practicality.

In a different approach, Yu et al. [14] proposed a GAN-based differential private image privacy protection framework for the Internet of Multimedia Things (IoMT). Their method identifies privacy-sensitive content in images using deep neural network techniques and protects it with synthetic content generated by GANs with differential privacy. The proposed framework was shown to effectively protect users' privacy while maintaining image utility.

Another significant contribution was made by Yang et al. [15] who proposed a facial image privacy protection method based on the principal components of adversarial segmented image blocks. Their method adds minor perturbation to the principal components of facial images to protect users' privacy and prevent distinct face-related features of the images from being easily extracted. The proposed method outperformed other similar methods in terms of generated image quality, operation speed, and target recognition network accuracy.

These studies highlight the ongoing efforts in the field of image privacy protection and provide a solid foundation for further research in this area [16].

Homomorphic encryption for image classification

Currently, Homomorphic Encryption (HE) has been employed in image classification tasks, but due to its high computational complexity, a balance usually needs to be struck between choosing datasets and ensuring classification accuracy.

In a study, the aim was to analyze data from The Cancer Genome Atlas (TCGA) dataset, which comprises 3,622 samples, covering 11 types of cancer, extracting genetic features from 25,128 genes. Through preprocessing methods, the number of genes was reduced to 4,096 or fewer, and a microAUC value of 0.9882 (85% accuracy) was achieved using a 1-layer shallow neural network [17].

Another study demonstrated that by optimizing the activation functions, the classification accuracy of CIFAR-10 could be improved by 4.27%. Moreover, pretraining the activation function optimization for Fashion-MNIST and CIFAR-10, even in different networks and datasets, could enhance the classification accuracy [18].

From these, it can be seen that homomorphic encryption has already achieved a certain level of accuracy in image classification tasks, though some optimization may be needed to enhance efficiency.

Methods

Preliminary preparation

Our proposed preliminary research and planning phase for constructing an encrypted classification model includes two main components: data preprocessing and building the initial Convolutional Neural Network (CNN) model [19]. Data preprocessing makes the data more suitable for analysis and modeling, and can improve the accuracy of the model. The subsequent image splitting CNN and encrypted fast CNN models for processing encrypted data need to be modified based on the initial CNN model.

Image preprocessing

This initial step is crucial for normalizing and enhancing the input data of the CNN model, thereby ensuring effective feature learning. We use the NEU-CLS dataset and perform a series of transformations [20].

1. Image Size Adjustment: Using the OpenCV library, we adjust all images to a uniform size of 112x112 pixels, thereby ensuring the consistency of the input data, which is a prerequisite for CNN models that require fixed-size input.
2. NEU-CLS Dataset: The NEU-CLS dataset of steel surface defects is a surface defect dataset that collects six typical surface defects of hot-rolled steel strip, namely rolling scale (RS), patches (Pa), cracks (Cr), pitted surface (PS), inclusions (In), and scratches (Sc). The dataset contains 1800 grayscale images, six different types of typical surface defects, each type of defect contains 300 samples. For defect detection tasks, the dataset provides annotations indicating the category and location of defects in each image [21].
3. Contrast Enhancement: We use histogram equalization to enhance image contrast, thereby improving the intensity distribution of the image, making hidden details more clearly visible, and improving the overall image quality. This stage is particularly important because it helps the model accurately distinguish features during the training process. The formula for histogram equalization is [22]:

$$H(v) = \sum_{i=0}^{v} h(i) \tag{1}$$

where $H(v)$ is the cumulative distribution function and $h(i)$is the histogram of pixel intensities.

4. Data Augmentation: To enhance the robustness of the model and prevent overfitting, we use data augmentation techniques, including rotation (±15°), horizontal flipping, and translation (±10%). Data augmentation mainly produces changes in the image, enriches the dataset, and allows our model to learn from a wider range of examples.

Original image classification model

Given the complexity of the NEU-CLS dataset, we construct a simple neural network for classification. Our aim is to validate the accuracy of classification after image block encryption, hence we opt for a Convolutional Neural Network (CNN) for training and testing, which sufficiently demonstrates our method. With fewer parameters, we can train an excellent model and effectively avoid overfitting [23].

1. Network Design The adjusted neural network targets encrypted data, starting with a convolutional layer with 7x7 convolution kernels, followed by an activation function layer and a fully connected layer with 256 input neurons and 128 output neurons. Finally, there is an activation function layer and an output layer with 128 input neurons and 6 output neurons. These features are combined for encrypted block image classification [24]. The calculation method of the convolutional layer is shown in Eq. 2:

$$(F * I)(i,j) = \sum_{m} \sum_{n} F(m,n)(i-m, j-n) \tag{2}$$

where, F is the filter or kernel, I is the input image, and (i, j) are the spatial coordinates.

2. Activation Function After the convolutional layer and pooling layer, the Rectified Linear Unit (ReLU) activation function is applied. ReLU introduces nonlinearity without affecting the receptive field of the convolutional layer. The Rectified Linear Unit (ReLU) activation function is represented as Eq. 3:

$$f(x) = max(0, x) \tag{3}$$

3. Optimization and Training We use cross-entropy as the loss function, which is the standard choice for classification problems. In terms of network optimization, we adopt the Adam optimizer, which can adaptively adjust the learning rate of each weight. The model is trained on the training set and validated on the validation set to evaluate its performance. The Adam optimizer updates the weights based on the adaptive estimation of the low-order moments [25]. The update rule of Adam can be represented as:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_{t+1}} + \epsilon} \hat{m}_{t+1} \tag{4}$$

where, θ is the parameter (weight), η is the learning rate, $\hat{v}_{t+1}$ and $\hat{m}_{t+1}$ are the estimates of the first and second moments of the gradient, and ϵ is a small constant to avoid division by zero.

Chunks classification model

Initially, we constructed a simple CNN model to process raw plaintext image data. However, to handle chunked image data, it was necessary to modify the original CNN model to accommodate chunked data [26].

1. Network Design The architecture of our CNN model was inspired by the classic design principles of convolutional networks [27]. Similar to the original CNN model, the adjusted neural network targets encrypted data, starting with a convolutional layer with 7x7 kernels, followed by a fully connected layer with 256 input neurons and 128 output neurons, and finally an activation function layer and a fully connected layer with 128 input neurons and 6 output neurons [28]. These features are combined to classify encrypted chunked images. The purpose of these layers is to learn hierarchical spatial features by applying filters to the input images and to capture increasingly complex patterns as the network deepens. They reduce the spatial size of the feature maps while retaining key information, thus lowering the computational complexity of the network and providing translational invariance. The fully connected layer is a major part of the CNN architecture, adjusted to accommodate 256 input neurons. This layer combines the high-level features learned in the preceding layers to perform the image classification task [29]. Finally, an output layer comprising six neurons is implemented, corresponding to the six categories in our dataset. Each neuron in this layer represents a different category, and the activation of these neurons provides a probability distribution of the possible categories for a given input image [30].
2. Activation Function After the convolutional and pooling layers, the Rectified Linear Unit (ReLU) activation function is applied, introducing non-linearity without affecting the receptive field of the convolutional layer [31]. The structure of the chunked convolutional neural network is shown in the figure:

3. Optimization and Training During network training, we use cross-entropy as the loss function [32]. Cross-entropy is a standard choice for classification problems as it measures the difference between predicted class probabilities and true class labels [33]. For network optimization, we employ the Adam optimizer, an algorithm capable of adaptively adjusting the learning rate of each weight in the network [34]. This optimizer combines the advantages of two other popular optimization algorithms, AdaGrad and RMSProp, providing an effective and efficient optimization method.

As shown in Fig. 4, the pre-classified chunk set is input into the modified adaptive chunk CNN model for training and testing. The classification results are then compared with the actual results.

The above data preprocessing and the construction steps of the original image CNN model and chunked image CNN model lay the foundation for subsequent homomorphic encryption and encrypted image classification stages.

Image chunking and encryption strategy

The second phase of our proposed method involves dividing the image into chunks and performing gradient calculations on these chunks. Homomorphic encryption (HE) is applied to image chunks that exceed a threshold value based on the calculation results, with the primary goal of protecting data privacy without hindering subsequent classification accuracy processes [35].

Image splitting

1. Image Splitting Each preprocessed image in the dataset is divided into small subregions of 28x28 pixels. This division simplifies the task of identifying different features in the image and helps to reduce the complexity of the problem [36].
2. Local Gradient Calculation Upon segmenting the image, we commence the computation of local gradients for each segment. This computation is executed utilizing the Sobel operator, a mature tool in image processing, capable of determining the edge direction and edge strength at every point within the image. These edge features are pivotal in image analysis and interpretation as they often correspond to significant boundaries within the image, such as the contours of objects [36]. The Sobel operator operates by taking the weighted difference of grayscale values of the pixels in the four neighboring regions (up, down, left, right) of each pixel in the image, reaching a maximum at the edges hence detecting them. It is a discrete differential operator used to approximate the gradient of the image intensity function. Applying this operator at any point within the image will generate the corresponding gradient vector or its normal vector. The Sobel operator used to calculate the local gradient can be represented as follows: For the x direction

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \tag{5}$$

For the y direction

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I \tag{6}$$

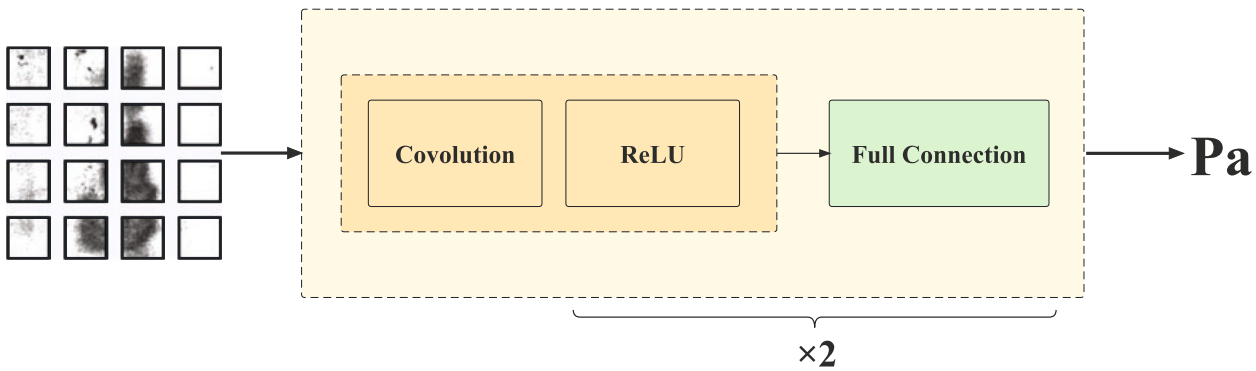

Fig. 4 Demonstration of the model structure performing convolutional operations on segments of the original image. The depicted convolutional network comprises one convolutional layer and two fully connected layers. The Rectified Linear Unit (ReLU) activation function is employed to introduce non-linearity, without impacting the receptive field of the convolutional layer

where G_x and G_y are the gradients in the x and y directions respectively, and I is the image. The star (*) represents the convolution operation. The formula for calculating the gradient magnitude is

$$G = \sqrt{G_x^2 + G_y^2} \tag{7}$$

3. Feature Information Recognition Following the computation of local gradients, we proceed to identify segments that encompass crucial feature information. We set the gradient threshold at 100, a value chosen based on multiple tests [37]. If a segment's gradient value surpasses this threshold, it is considered to contain ample feature information; otherwise, it is deemed to lack critical feature information. This criterion aids us in distinguishing segments with significant features from those without, allowing us to concentrate computational resources on segments that are more likely to contribute to the overall image classification [38]. The entire chunking and gradient computation process above is shown in Fig. 5.
4. Chunk Indexing Finally, assign an index to each chunk. Chunks identified as containing feature information are given a positive index, while chunks that do not contain feature information are given a negative index. This indexing scheme helps track split chunks and their related information, providing a systematic method for managing the large amount of data generated during the image splitting process. It also aids subsequent steps in our method, such as encrypting feature-rich split chunks and inputting these split chunks into our improved ciphertext convolutional neural network classification model [39, 40].

Chunks encryption

1. Encrypting Feature-Rich Image Chunks Once chunks containing important feature information are identified, we begin encrypting these fragments using the fully homomorphic Cheon-Kim-Kim-Song (CKKS) scheme. CKKS is a form of homomorphic encryption that allows complex operations to be performed on encrypted data. This feature is particularly beneficial for computations involving image data, as image data often involves non-linear operations such as convolution. By using the CKKS scheme, CKKS can directly encode, encrypt, and operate on double-precision floating-point real numbers or even complex numbers, thus protecting the privacy of image data while still performing meaningful computations [41].
2. Encryption Process Parameter Selection: Choosing appropriate parameters is crucial, including the degree of the polynomial, the modulus, and the error distribution, among others. These parameters impact the security and efficiency of the encryption scheme [29]. When using multiple polynomial moduli, different sizes of images require different sizes of keys, as shown in Table 1.

Key Generation: Public and Private Key Generation: A random private key is generated, and the public key is computed using the private key. The private key is typically a short vector, while the public key is a matrix formed by combining the private key with some random elements [42].

Encryption: Message Encoding: The message to be encrypted is encoded into a polynomial. This involves mapping the message to a complex domain or integer ring and includes some scaling and adjustment

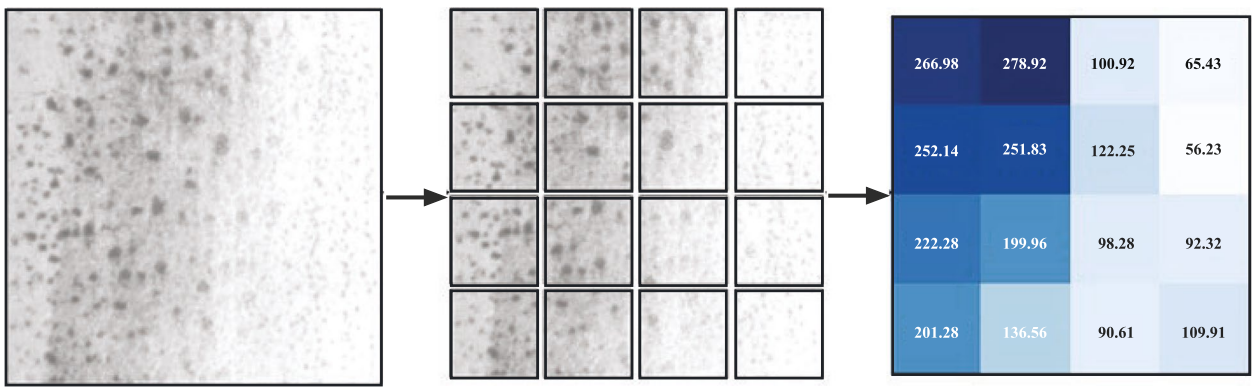

Fig. 5 Illustration of the process of filtering chunks based on feature values. The value signifies the content of features, with chunks containing a greater number of features exhibiting larger values. Chunks that surpass a predetermined threshold are deemed to contain feature chunks

Table 1 Key size required for two sizes of images using different polynomial modules

Image size	polynomial modulus	All keys	Public key	Secret key	Galois keys	Relin keys
112*112	32768	6.17 MB	4.12 MB	2.06 MB	1.69 GB	61.79 MB
28*28	16384	3.09 MB	2.06 MB	1.03 MB	805.98 MB	30.93 MB

Note: There is a strong correlation between polynomial modulus and image size and key size. When using multiple polynomial moduli, different sized images require different sized keys

operations to maintain precision. Actual Encryption: The public key and the encoded message, along with some random values, are used to create the encrypted ciphertext. The encryption process in the CKKS scheme involves polynomial multiplication and modulus reduction operations [43], as illustrated in Fig. 6.

Computation: Computations are performed on the ciphertext. These computations are executed on the encrypted data, without the need for decryption. In the CKKS scheme, complex polynomial computations can be carried out, enabling more complex operations to be performed on encrypted data (Fig. 7).

Decryption: The private key is used to decrypt the encrypted ciphertext to recover the original message [44]. In the CKKS scheme, the decryption process involves polynomial multiplication and modulus reduction operations. Decryption is only performed after the necessary computations on the encrypted data have been executed, ensuring that the original image data is revealed only when necessary.

Message Decoding: The decrypted polynomial is decoded back into the original message. This involves inverse mapping operations, as well as some adjustments and scaling to restore the original precision.

The CKKS and encryption scheme we use in our method is complex and involves various mathematical operations. A simplified representation of the encryption process is as follows:

$$E(m) = (m + r) \mod q \tag{8}$$

where $E(m)$ is the encrypted message, m is the original message, r is a random number, and q is a large prime number.

Chunks merge and storage

1. Image Chunk Merging After encryption, the feature and non-feature fragments are merged into block groups [45]. The order of these block groups reflects the order of the original images, with each block group corresponding to an original image.

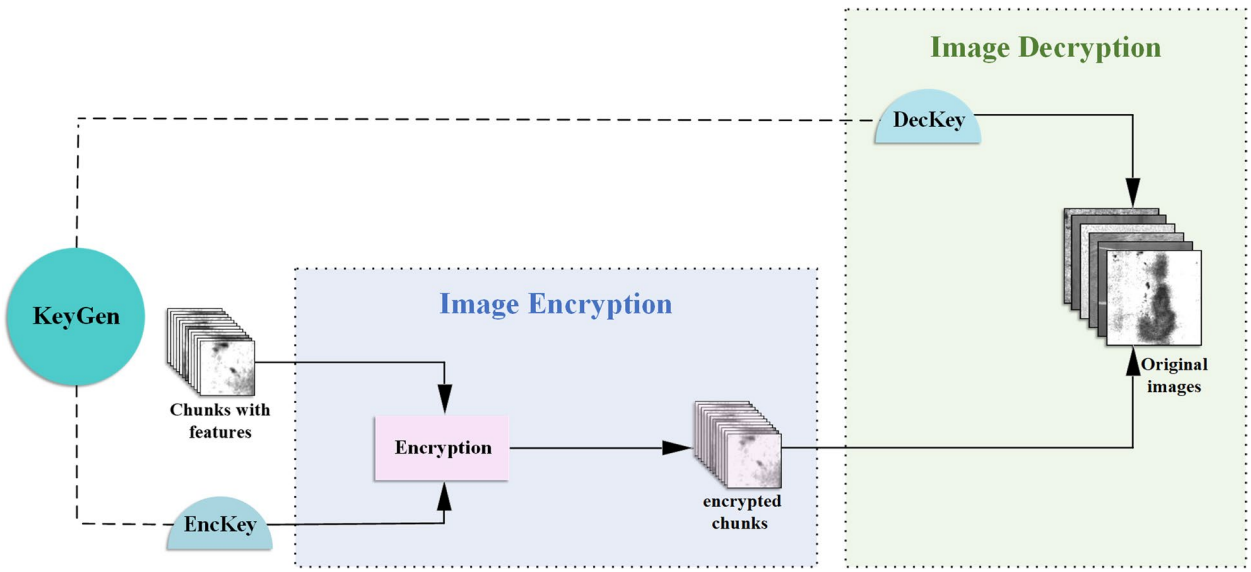

Fig. 6 Representation of the application of full homomorphic encryption in image processing. The process initiates with the KeyGen function generating an encryption key (EncKey) and a decryption key (DecKey). The EncKey is utilized to encrypt image chunks with distinctive features, while the DecKey is ultimately employed to decrypt these encrypted chunks

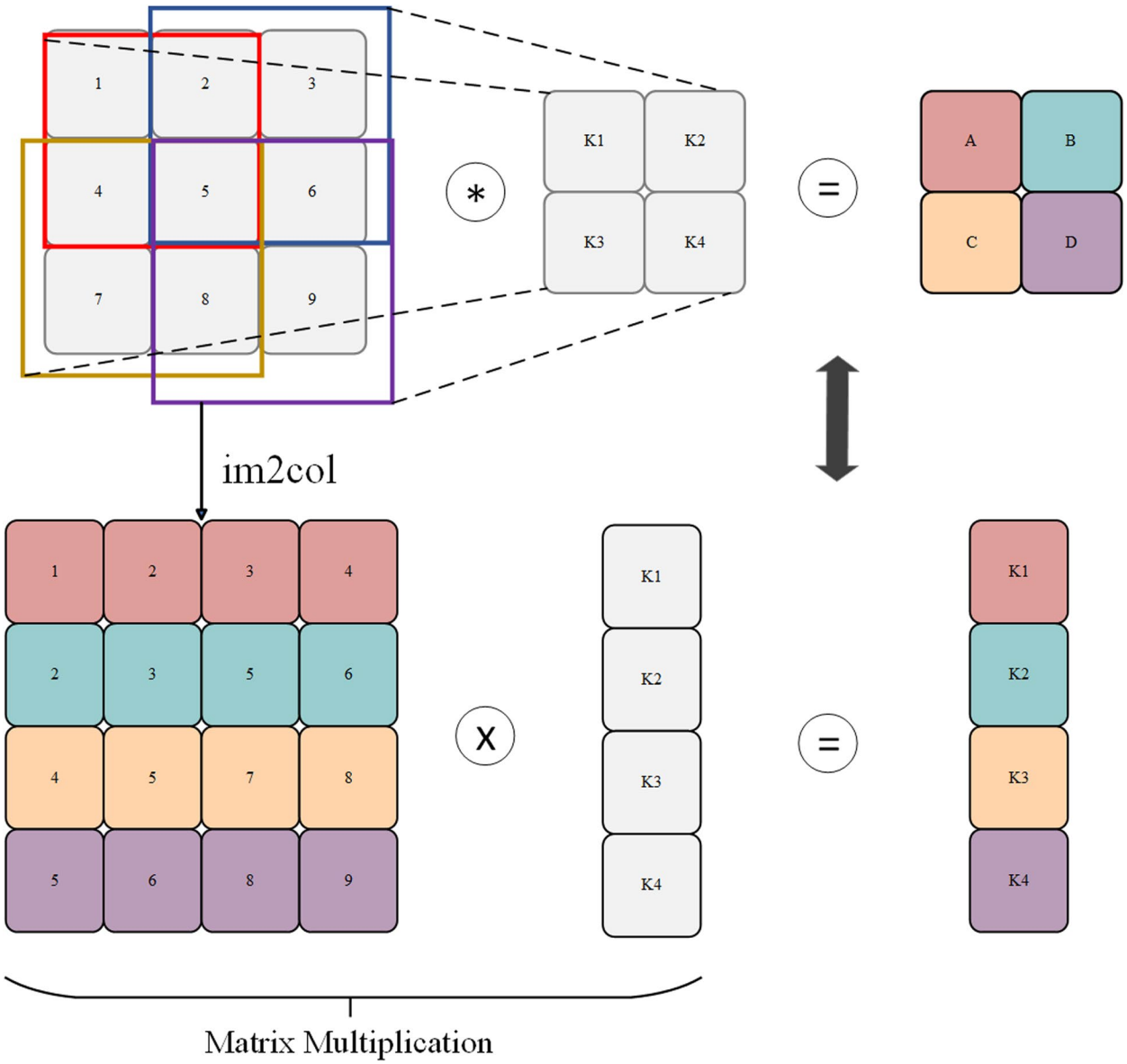

Fig. 7 Illustration of the process of subjecting the encrypted image to a convolution operation. The model performs an image-to-column convolution, effectively transforming the 2D convolution into a singular matrix multiplication operation

2. Block Group Storage Once the block groups are formed, they are stored in arrays or lists. This storage format is chosen because in subsequent stages of our method, it is necessary to conveniently access and manipulate these block groups. Arrays and lists allow efficient access to their elements and allow easy manipulation of data, making them ideal for our purposes. By storing block groups in this way, we can easily retrieve and process encrypted image data in the later stages of our method [46].

Through this process, we prepare the dataset for subsequent steps using homomorphic encryption technology, thereby ensuring data privacy. These steps include modifying the convolutional neural network (CNN) model to handle encrypted data and training this modified model on the encrypted dataset. By performing these operations on encrypted data, we can maintain the privacy of image data while obtaining accurate image classification results. This balance between data privacy and computational efficiency is a key aspect of our method and

demonstrates the potential of homomorphic encryption in privacy-protecting image classification tasks.

Encrypted chunks classification model

The final stage of our method involves training an adjusted Convolutional Neural Network (CNN) on encrypted data and evaluating its performance. In this stage, we modify the image chunking CNN model to handle encrypted data and assess the efficiency and effectiveness of our method.

Classification model modification

1. Homomorphic encryption operations Replace the original layers (convolutional layer, pooling layer, and fully connected layer) of the CNN model with homomorphic encryption operations that can handle encrypted data [47, 48]. The adjusted neural network targets encrypted data, starting with a convolutional layer with 7x7 convolution kernels, followed by a fully connected layer with 256 input neurons and 128 output neurons. These features are combined for encrypted chunk image classification. Finally, an output layer with 6 neurons is implemented, corresponding to the 6 categories in our dataset. The structure of the chunked convolutional neural network is shown in the Fig. 8:
2. Activation and loss functions Current homomorphic encryption schemes require careful selection of the activation function and loss function for the neural network. In our method, due to the complexity of the NEU-CLS dataset, we use the square activation function and cross-entropy loss function to be compatible with encryption technology.

$$y = x^2 \tag{9}$$

The cross-entropy loss function for a binary classification problem can be defined as:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{10}$$

where y is the true label, $\hat{y}$ is the predicted label, and N is the number of samples.

Training and testing with encrypted chunks

1. Testing with encrypted data After the model is trained, we test its performance on the encrypted test set. The testing process includes inputting the encrypted image data into the trained model and comparing the model's prediction results with the true labels of the images. Due to the complexity of homomorphic encryption operations, this testing process takes longer than testing with unencrypted data [17]. However, our scheme reduces the encryption computation time without losing accuracy. The testing process rigorously evaluates the performance of our model, allowing us to assess the degree of generalization of our model to unseen data. This generalization performance is an important aspect of any machine learning model, as it indicates the model's ability to make accurate predictions on new data, not limited to the specific examples it was trained on [49].
2. Recording performance metrics After the testing process, we record key performance metrics such as accuracy and loss value. The accuracy metric provides the proportion of images correctly classified by the model, giving us a direct understanding of the model's performance. On the other hand, the loss value can measure the confidence of the model in its predictions. The smaller the loss value, the more confident the model is in its predictions, which usually means better performance [50].

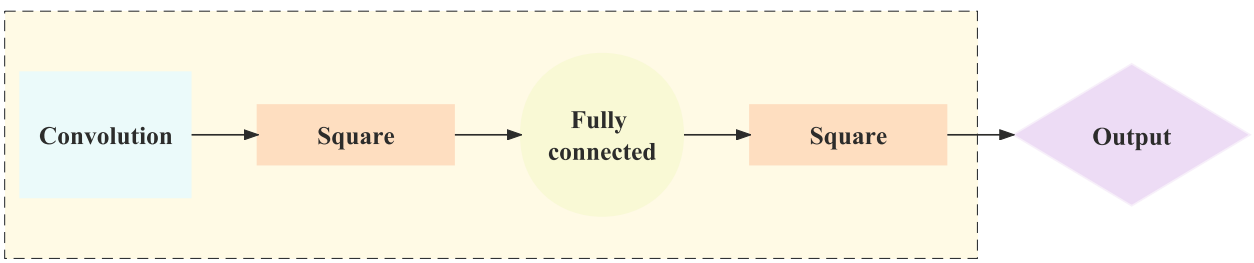

Fig. 8 Depiction of the model structure executing convolutional operations on image chunks. The illustrated convolutional network comprises a convolutional layer, a 2-layer activation function using a square function to introduce non-linearity, and two fully connected layers. The square function is employed to introduce non-linearity without impacting the receptive field of the convolutional layer

These metrics provide valuable information for understanding the performance of the model and the effectiveness of our method. By recording these metrics, we can track the progress of the model over time, identify areas that need improvement, and compare our method with other methods. This recording process is an important part of our methodology, as it allows us to critically evaluate our method and continually strive for improvement [30].

Classification results and performance evaluation

1. Performance metric comparison We compare the performance metrics-accuracy and loss value-of the original data and encrypted data. This comparison helps to understand the impact of encryption on model performance:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}. \tag{11}$$

   By comparing the accuracy and loss values when the model is trained on original data and encrypted data, we can gain a deeper understanding of the trade-offs involved in privacy-preserving image classification using homomorphic encryption technology.
2. Encryption impact analysis In addition, we conduct a comprehensive analysis of the impact of encryption operations on model performance and computational complexity. This includes evaluating the impact of homomorphic encryption operations on the model's testing time and the model's accuracy. By studying these impacts, we can better understand the computational costs associated with our privacy-preserving method and identify potential areas for optimization.
3. Privacy protection evaluation Based on experimental results, we evaluate the effectiveness of our method in protecting data privacy [51]. This evaluation is crucial for confirming whether our method has achieved the dual goals of data privacy protection and efficient image classification. By demonstrating that our method can maintain high classification accuracy while protecting data privacy, we can validate the effectiveness of our method. Through these comprehensive steps, we aim to leverage the capabilities of deep learning and homomorphic encryption to design a robust and privacy-preserving image classification method. Our method demonstrates the potential of combining advanced encryption technology with deep learning models to achieve secure and efficient image classification [52].

Result and discussion

Preliminary work for encrypted chunks classification model

We performed a series of preprocessing steps on images from the NEU-CLS dataset. These steps include resizing the images to a uniform size, converting the images to grayscale to reduce computational complexity, applying histogram equalization to enhance image contrast, and implementing data augmentation techniques to increase the diversity of the dataset [53]. After the preprocessing stage, we constructed a Convolutional Neural Network (CNN) model. This model includes two convolutional layers, two pooling layers, two fully connected layers, and an output layer. We used Rectified Linear Units (ReLU) as the activation function to introduce non-linearity into the model and used the Adam optimizer for efficient parameter updates. This model laid a solid foundation for our subsequent modifications and adaptation to encrypted image chunking computations, adopting a simple neural network to better adapt to our computations on complex datasets.

Modification of encrypted chunks classification model

We adjusted the original CNN model to handle encrypted chunk image data [54]. This adjustment involves modifying the input of the fully connected layer to adapt to the changes brought about by encryption operations. This adjusted model aims to handle chunked and encrypted image data, allowing us to maintain data privacy while performing image classification [55]. The encrypted image chunking convolutional neural network we constructed can be applied to various datasets, and image classification of various encrypted image chunks can be achieved by simply adjusting the input and output of the fully connected layer.

Image splitting and chunks encryption

After adjusting the model, we split each image into subregions of 28x26 pixels. Then, we calculated the local gradients of each subregion using the Sobel operator, a common technique for edge detection in image processing. Based on the calculated gradient values, we divided the regions into areas with or without important features. Then, we used the Complex Key-Value Storage (CKKS) homomorphic encryption scheme to homomorphically encrypt the areas with important features. After encryption, the regions were merged into chunk groups, maintaining their original order to preserve the spatial relationships between different parts of the image. In the analysis process, we found that the time required to encrypt only the chunks containing features using our complete method was the shortest compared to the

time required to encrypt the entire image using a generic model, encrypt all chunks using our incomplete method, and encrypt only the chunks containing features using our complete method, as shown in Table 2.

Results of classification and model performance

The convolutional layer, pooling layer, and fully connected layer of the original CNN model were replaced with homomorphic encryption operations that can handle encrypted data. This transformation required the use of a square activation function and cross-entropy loss function compatible with the CKKS scheme. Then, we retrained the adjusted network with the encrypted training set. We validated the model on the encrypted validation set, thereby evaluating the model's performance on unseen data.

During the encryption process, we chunked according to the combination of image chunk groups. The encryption speed of this combination is on average 0.1 seconds faster than the original image without chunking, showing a significant time advantage, especially on large datasets.

Table 2 Time taken to encrypt a whole image using the generic model, to encrypt all chunks using our incomplete method, and to encrypt only chunks containing features using our full method

Method	Number of sheets/chunks	Image size	time(ms)
General Model	1	112*112	484
Ours all chunks	16	28*28	345
Ours (full method)	n	28*28	235

The classification results of the original image chunking model and the adapted encrypted model are shown in the attached Fig. 9. The overall classification accuracy after encryption has significantly improved compared to the original image classification accuracy.

The pre-classified chunk groups are input into the modified adaptive chunking CNN model for testing, and the classification results are compared with the actual results, as shown in Fig. 10a. The pre-classified encrypted chunk groups are input into the modified adaptive encrypted chunking CNN model for testing, and the classification results are compared with the actual results, as shown in Fig. 10b.

Comparing our method with existing research shows the superiority of the accuracy of our method. The comparison results are shown in Table 3.

Finally, we compared the performance of models trained on original data and encrypted data. The focus of the comparison was on accuracy, loss value, and the impact of encryption operations on model performance and computational complexity. We also evaluated the effectiveness of the encrypted neural network in data privacy protection.

Summary

Our research presents a novel approach to image classification that maintains data privacy by utilizing encrypted image chunks and an adjusted Convolutional Neural

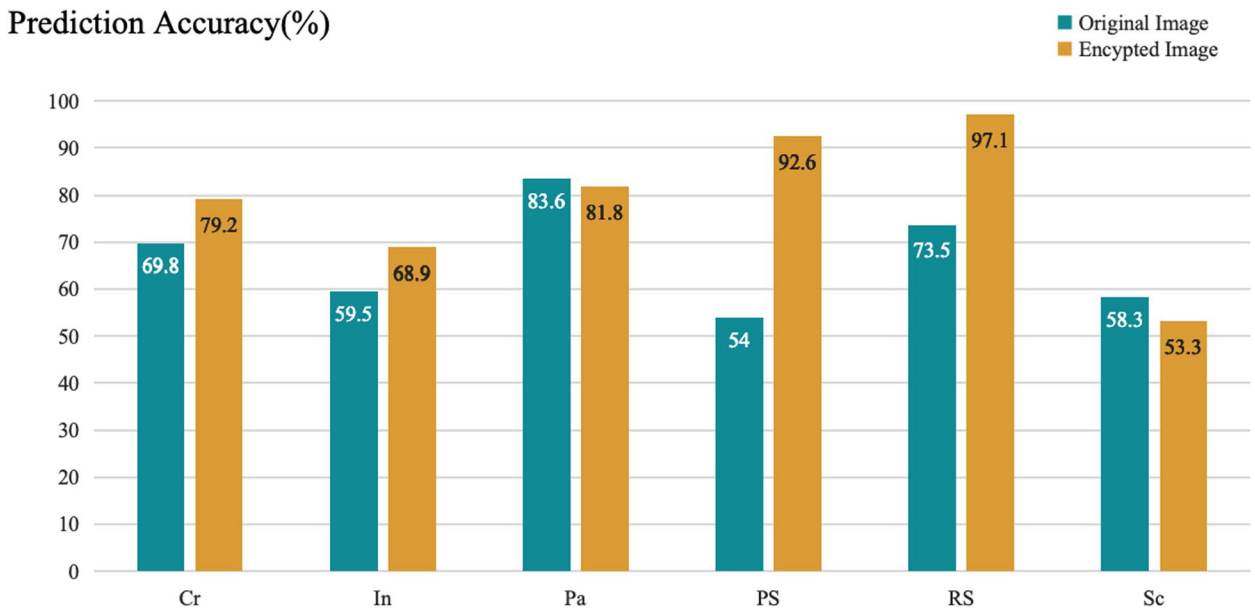

Fig. 9 Illustration of the comparative results of the original image chunking model and the adaptive encryption model for classification. The depicted values represent the accuracy of original image classification and encrypted image chunk classification for the six NEU-CLS images, respectively

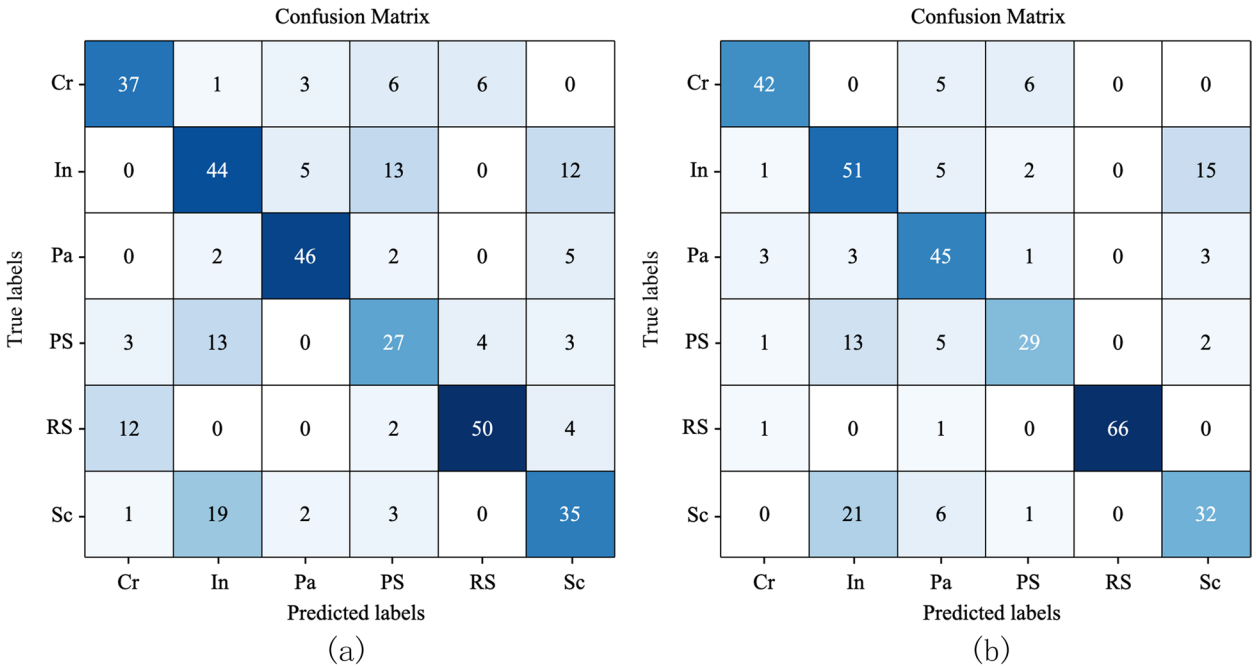

Fig. 10 Comparative representation of evaluation results from the plaintext model and the chunked ciphertext model. The diagonal elements represent the test accuracy values. **a** Illustrates the outcomes of the classification test conducted on the plaintext dataset, juxtaposed with the true classification results. **b** Depicts the outcomes of the classification test conducted on the chunked ciphertext dataset, compared with the true classification results

Table 3 The accuracy comparison of the method [17] and the optimized activation function method [18] with our method for analyzing The Cancer Genome Atlas (TCGA) data set is shown

Method	Dataset	Accuracy(%)	Activation Function
Multi-label Tumor	TCGA	85	softmax
Mish Fuction	Fashion-MNIST	84.68	Approximated Mish
Mish Fuction	CIFAR-10	70.2	Approximated Mish
Our Method	NEU-CLS	97.1	Square

Network (CNN). The study is divided into four sections, each detailing a specific stage of our method and its results.

In the initial stage, we preprocessed images from the NEU-CLS dataset, which included resizing, grayscale conversion, histogram equalization, and data augmentation. A simple CNN model was then constructed, laying the groundwork for subsequent modifications to handle encrypted image chunks.

The second stage involved adjusting the original CNN to handle encrypted chunked image data. This adjustment mainly involved modifying the inputs of the fully connected layer to accommodate changes brought about by encryption operations. The adjusted model is designed to handle chunked and encrypted image data, enabling image classification while preserving data privacy.

In the third stage, we segmented each image into 28x26 pixel subregions and calculated local gradients using the Sobel operator. Based on the computed gradient values, we divided the regions into those with or without significant features. Homomorphic encryption was then applied to regions with significant features using the Complex-Valued Cheon-Kim-Kim-Song (CKKS) scheme. The encrypted regions were merged into chunk groups, maintaining their original order to preserve spatial relationships between different parts of the image.

In the final stage, we replaced the convolutional layer, pooling layer, and fully connected layer of the original CNN model with homomorphic encryption operations compatible with the CKKS scheme. The adjusted network was then retrained with the encrypted training set and validated on the encrypted validation set.

Our results showed that the time required for encrypting only the feature-containing chunks using our complete method was the shortest compared to encrypting the entire image using a generic model or encrypting all chunks using our incomplete method. Furthermore, the performance of the models trained on original and encrypted data was compared, focusing on accuracy, loss values, and the impact of encryption operations on model performance and computational complexity. The effectiveness of the encrypted neural network in data privacy protection was

also evaluated. Our method demonstrated significant time advantage, especially on large datasets, and maintained high classification accuracy while protecting data privacy.

Conclusion

In this research, we presented a pioneering approach to image classification that seamlessly integrates accuracy and data privacy by utilizing encrypted image chunks and an adjusted Convolutional Neural Network (CNN). Our method demonstrated significant time advantages, especially on large datasets, and maintained high classification accuracy. The emphasis on encrypting only the feature-rich chunks not only ensures data privacy but also offers computational benefits. The flexibility of the CNN model in accommodating encrypted data structures underscores its potential in evolving data privacy landscapes. Future research directions include further optimization of the CNN model, exploration of alternative encryption schemes, and application to diverse datasets. The findings of this study have broad implications for secure and efficient image processing across various domains.

Acknowledgements
We would like to express our sincere thanks to the editors and reviewers for their insightful comments and suggestions on improving this paper.

Authors' contributions
All authors were involved in the research for this paper. Xiaoyu Li and Huixue Jia led the entire work. Cong Wang carried out the detailed research, including background study and review compilation. Huixue Jia and Jie Yang were responsible for the modeling and experimental analysis part of the whole paper. Huixue Jia and Daomeng Cai were responsible for writing the thesis as well as the experimental evaluation. Weidong Qian and Shan Yang was responsible for the correction and layout of the thesis. All authors read and approved the final draft.

Funding
This work was supported in part by the Opening Project of Intelligent Policing Key Laboratory of Sichuan Province (No.ZNJW2023KFMS005) in part by the National Key R\&D Program of China (No.J2019-V-0001-0092) and in part by the Science and Technology Projects of the Ministry of Public Security of China (No.2022JSM04).

Declarations

Ethics approval and consent to participate
This article does not contain any studies with human participants or animals performed by any of the authors.

Competing interests
The authors declare no competing interests.

Received: 30 August 2023 Accepted: 28 October 2023
Published online: 12 December 2023

References

1. Wang F, Li G, Wang Y, Rafique W, Khosravi MR, Liu G, Liu Y, Qi L (2023) Privacy-aware traffic flow prediction based on multi-party sensor data with zero trust in smart city. ACM Trans Internet Technol 23(3):1–19
2. Zheng Y, Li Z, Xu X, Zhao Q (2022) Dynamic defenses in cyber security: Techniques, methods and challenges. Digit Commun Netw 8(4):422–435
3. Gayathri S, Gowri S (2023) Securing medical image privacy in cloud using deep learning network. J Cloud Comput 12(1):40
4. Yang Y, et al (2023) ASTREAM: Data-Stream-Driven Scalable Anomaly Detection With Accuracy Guarantee in IIoT Environment. IEEE Trans Netw Sci Eng 10(5):3007–3016. https://doi.org/10.1109/TNSE.2022.3157730
5. Lei W, Zhou Y, Lin X (2021) A physical layer security scheme for full-duplex communication systems with residual self-interference and non-eavesdropping CSI. Digit Commun Netw 7(3):352–361. https://doi.org/10.1016/j.dcan.2020.07.004
6. Chen Z, Cheng G, Xu Z, Guo S, Zhou Y, Zhao Y (2022) Length matters: Scalable fast encrypted internet traffic service classification based on multiple protocol data unit length sequence with composite deep learning. Digit Commun Netw 8(3):289–302
7. Saharan S, Laxmi V, Bezawada B, Gaur MS (2021) Scaling & fuzzing: Personal image privacy from automated attacks in mobile cloud computing. J Inf Secur Appl 60:102850
8. Ibtihal M, Driss EO, Hassan N (2020) Homomorphic Encryption as a Service for Outsourced Images in Mobile Cloud Computing Environment. In I. Management Association (Ed.), Cryptography: Breakthroughs in Research and Practice (pp. 316–330). IGI Global. https://doi.org/10.4018/978-1-7998-1763-5.ch019
9. Geng Y et al (2019) Homomorphic encryption technology for cloud computing. Procedia Comput Sci 154:73–83
10. Wang J, Wu F, Zhang T, Wu X (2022) DPP: Data Privacy-Preserving for Cloud Computing based on Homomorphic Encryption. 2022 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Suzhou, China, pp. 29–32. https://doi.org/10.1109/CyberC55534.2022.00016
11. Dai H, et al (2023) Bloom Filter With Noisy Coding Framework for Multi-Set Membership Testing. IEEE Trans Knowl Data Eng 35(7):6710–6724. https://doi.org/10.1109/TKDE.2022.3199646
12. Qi L, Lin W, Zhang X, Dou W, Xu X, Chen J (2023) A Correlation Graph Based Approach for Personalized and Compatible Web APIs Recommendation in Mobile APP Development. IEEE Trans Knowl Data Eng 35(6):5444–5457. https://doi.org/10.1109/TKDE.2022.3168611
13. Yang J, Liu J, Han R, Wu J (2021) Transferable face image privacy protection based on federated learning and ensemble models. Complex Intell Syst 7(5):2299–2315
14. Yu J, Xue H, Liu B, Wang Y, Zhu S, Ding M (2020) Gan-based differential private image privacy protection framework for the internet of multimedia things. Sensors 21(1):58
15. Yang J, Liu J, Wu J (2020) Facial image privacy protection based on principal components of adversarial segmented image blocks. IEEE Access 8:103385–103394
16. Wu S, Shen S, Xu X, Chen Y, Zhou X, Liu D, Xue X, Qi L (2022) Popularity-aware and diverse web apis recommendation based on correlation graph. IEEE Trans Comput Soc Syst 10(2):771–782
17. Hong S, Park JH, Cho W, Choe H, Cheon JH (2022) Secure tumor classification by shallow neural network using homomorphic encryption. BMC Genomics 23(1):1–19
18. Yagyu K, Takeuchi R, Nishigaki M, Ohki T (2023) Improving Classification Accuracy by Optimizing Activation Function for Convolutional Neural Network on Homomorphic Encryption. In: Barolli, L. (eds) Advances on Broad-Band Wireless Computing, Communication and Applications. BWCCA 2022. Lecture Notes in Networks and Systems, vol 570. Springer, Cham. https://doi.org/10.1007/978-3-031-20029-8_10
19. Tang X, Zheng D, Kebede GS, Li Z, Li X, Lu C, Li L, Zhou Y, Yang S (2023) An automatic segmentation framework of quasi-periodic time series through graph structure. Applied Intelligence 1–18
20. Zheng D, Ran Z, Liu Z, Li L, Tian L (2020) An efficient bar code image recognition algorithm for sorting system. Comput Mater Contin 64(3):1885–1895
21. Feng X, Gao X, Luo L (2021) X-sdd: A new benchmark for hot rolled steel strip surface defects detection. Symmetry 13(4):706
22. Tanaka H, Taguchi A (2023) Brightness preserving generalized histogram equalization with high contrast enhancement ability. IEICE Trans Fundam Electron Commun Comput Sci 106(3):471–480
23. Chen HC, Widodo AM, Wisnujati A, Rahaman M, Lin JCW, Chen L, Weng CE (2022) Alexnet convolutional neural network for disease detection and classification of tomato leaf. Electronics 11(6):951

24. Putra IP, Rusbandi R, Alamsyah D (2022) Klasifikasi penyakit daun jagung menggunakan metode convolutional neural network. Jurnal Algoritme 2(2):102–112
25. Rajendar S, Kaliappan V (2022) Sensor Data Based Anomaly Detection in Autonomous Vehicles using Modified Convolutional Neural Network. Intell Autom Soft Comput 32:859–875. https://doi.org/10.32604/iasc.2022.020936
26. Qian J, Zhang P, Zhu H, Liu M, Wang J, Ma X (2023) Lhdnn: Maintaining high precision and low latency inference of deep neural networks on encrypted data. Appl Sci 13(8):4815
27. Vieira N (2023) Bicomplex neural networks with hypergeometric activation functions. Adv Appl Clifford Algebras 33(2):20
28. Ouma YO, Omai L (2023) Flood Susceptibility Mapping Using Image-Based 2D-CNN Deep Learning: Overview and Case Study Application Using Multiparametric Spatial Data in Data-Scarce Urban Environments. Int J Intell Syst 2023(5672401):23. https://doi.org/10.1155/2023/5672401
29. Lee JW, Kang H, Lee Y, Choi W, Eom J, Deryabin M, Lee E, Lee J, Yoo D, Kim YS et al (2022) Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. IEEE Access 10:30039–30054
30. Ma'arif A, Rahmaniar W, Fathurrahman HIK, Frisky AZK, et al (2022) Understanding of convolutional neural network (cnn): A review. Int J Robot Control Syst 2(4)
31. Polyakova M (2023) Image segmentation with a convolutional neural network without pooling layers in dermatological disease diagnostics systems. Radio Electron Comput Sci Control 1:51–51
32. Huang J, Niu G, Guan H, Song S (2023) Ultra-short-term wind power prediction based on lstm with loss shrinkage adam. Energies 16(9):3789
33. Barbu A (2023) Training a two-layer relu network analytically. Sensors 23(8):4072
34. Xie Z, Shu C, Fu Y, Zhou J, Chen D (2023) Balanced loss function for accurate surface defect segmentation. Appl Sci 13(2):826
35. Wang F, Zhu H, Srivastava G, Li S, Khosravi MR, Qi L (2021) Robust collaborative filtering recommendation with user-item-trust records. IEEE Trans Comput Soc Syst 9(4):986–996
36. Jain N, Nandakumar K, Ratha N, Pankanti S, Kumar U (2021) Efficient cnn building blocks for encrypted data. arXiv preprint arXiv:2102.00319
37. Hou Z, Lv K, Gong X, Wan Y (2023) A remote sensing image fusion method combining low-level visual features and parameter-adaptive dual-channel pulse-coupled neural network. Remote Sens 15(2):344
38. Xie T, Yamana H, Mori T (2022) Che: Channel-wise homomorphic encryption for ciphertext inference in convolutional neural network. IEEE Access 10:107446–107458
39. Tiwari PK, Kannan K, Veeraiah D, Ranjan N, Singh J, Alshammri GH, Halifa A (2022) Security Protection Mechanism in Cloud Computing Authorization Model Using Machine Learning Techniques. Wirel Commun Mob Comput 2022(1907511):12. https://doi.org/10.1155/2022/1907511
40. Han D, Tian M, Gong C, Zhang S, Ji Y, Du X, Wei Y, Chen L (2022) Image classification of forage grasses on etuoke banner using edge autoencoder network. PLoS ONE 17(6):e0259783
41. Rahulamathavan Y (2022) Privacy-preserving similarity calculation of speaker features using fully homomorphic encryption. arXiv preprint arXiv:2202.07994
42. Pathak V (2022) Lattices, homomorphic encryption, and ckks. arXiv preprint arXiv:2205.03511
43. Su Y, Tu Z, Wang X, Lin C (2023) A secure face recognition scheme based on CKKS homomorphic encryption and neural network. Journal of Lanzhou University of Technology 49(2):103–109
44. El-shafai W, El-Hag NA, El-Banby GM, Khalaf AA, Soliman NF, Algarni AD, El-Samie FE (2021) An Efficient CNN-Based Automated Diagnosis Framework from COVID-19 CT Images. Computers, Materials & Continua
45. Chen X, Gong M, Gan Z, Lu Y, Chai X, He X (2023) Cie-lscp: color image encryption scheme based on the lifting scheme and cross-component permutation. Complex Intell Syst 9(1):927–950
46. Mabruri AS et al (2020) Data security system of text messaging based on android mobile devices using advanced encrytion standard dynamic s-box. J Soft Comput Explor 1(1):39–46
47. Chiang J (2022) Volley revolver: A novel matrix-encoding method for privacy-preserving neural networks (inference). arXiv preprint arXiv:2201.12577
48. Joshi B, Joshi B, Mishra A, Arya V, Gupta AK, Peraković D (2022) A comparative study of privacy-preserving homomorphic encryption techniques in cloud computing. Int J Cloud Appl Comput (IJCAC) 12(1):1–11
49. Gómez-Jemes L, Oprescu AM, Chimenea-Toscano Á, García-Díaz L, Romero-Ternero MdC (2022) Machine learning to predict pre-eclampsia and intrauterine growth restriction in pregnant women. Electronics 11(19):3240
50. Liu L, Wang M, Li G, Wang Q (2022) Construction of Predictive Model for Type 2 Diabetic Retinopathy Based on Extreme Learning Machine. Diabetes Metab Syndr Obes 24;15:2607–2617. https://doi.org/10.2147/DMSO.S374767
51. Sirisha U, Chandana BS (2023) Privacy preserving image encryption with optimal deep transfer learning based accident severity classification model. Sensors 23(1):519
52. Kalapaaking AP, Stephanie V, Khalil I, Atiquzzaman M, Yi X, Almashor M (2022) Smpc-based federated learning for 6g-enabled internet of medical things. IEEE Netw 36(4):182–189
53. Yang Y, Ding S, Liu Y, Meng S, Chi X, Ma R, Yan C (2022) Fast wireless sensor for anomaly detection based on data stream in an edge-computing-enabled smart greenhouse. Digit Commun Netw 8(4):498–507
54. Almagrabi AO, Bashir A (2022) A classification-based privacy-preserving decision-making for secure data sharing in internet of things assisted applications. Digit Commun Netw 8(4):436–445
55. Kong L, Wang L, Gong W, et al (2022) LSH-aware multitype health data prediction with privacy preservation in edge environment. World Wide Web 25:1793–1808. https://doi.org/10.1007/s11280-021-00941-z

Publisher's Note