PRIVACY-PRESERVING IMAGE PROCESSING FOR FACIAL RECOGNITION

Amaan Vora, Jahnavi Shah
Ethical Statistics
Department of Statistics
Rutgers University, New Brunswick
New Jersey, 08901
av860@scarletmail.rutgers.edu, js3249@scarletmail.rutgers.edu

ABSTRACT

Facial Recognition is an increasingly employed tactic to recognize people around the world. With multiple applications in many professions, it is clear that this technology is here to stay. However, there are quite a few concerns with the whole process - from data procurement to accuracy issues. Through this research paper, we aim to address the critical issue of privacy in facial recognition systems by exploring and implementing advanced privacy-preserving methods. We investigate the applicability of techniques such as differential privacy and homomorphic encryption to safeguard sensitive facial biometric data during the recognition process. Our primary focus is minimizing the risk of re-identification, a pivotal concern in deploying biometric technologies. Additionally, we develop a robust facial recognition system with real-time capabilities, ensuring its performance across varying lighting conditions, angles, and facial expressions. The accuracy of the proposed system is rigorously evaluated using standard benchmarks. We also ensure that with this simplistic simulation of a large-scale system, we can make this model accessible for all conventional devices, making it easier for this privacy-preserving technology to perform on all devices.

1 INTRODUCTION

The ubiquity of facial recognition technology has revolutionized identity verification, providing a convenient means of confirming individuals based on their distinct facial features. However, the widespread adoption of biometrics, particularly facial recognition, has given rise to privacy apprehensions. As these technologies become more pervasive, the imperative for robust privacy protection solutions becomes increasingly pronounced.

Initially, facial recognition systems relied on centralized servers for biometric matching, raising concerns about individual privacy. These apprehensions spurred investigations into Privacy-Enhancing Technologies (PETs) as potential remedies. Early endeavors, such as secure multiparty computation, sought to facilitate matching without exposing sensitive data to untrusted entities. Although progress has been made, persistent challenges have prompted a continual exploration for more effective privacy protection solutions.

Researchers have delved into various technologies for privacy-preserving methods in facial recognition. Differential privacy, which ensures that including or excluding an individual's data doesn't significantly impact calculation outcomes, has emerged as a focal point. Another avenue of exploration is homomorphic encryption, enabling computations on encrypted data without decryption. While these technologies exhibit promise in preserving biometric information during matching, the challenge lies in seamlessly integrating them into real-world facial recognition systems.

This project introduces a pioneering approach to privacy-preserving image processing for facial recognition, with a primary emphasis on harnessing the power of homomorphic encryption. Our proposed system strategically incorporates secure multiparty computation techniques to adeptly conceal both biometrics and matching results from the server, thereby ensuring the complete non-disclosure of the input image and recognition outcome. This innovative methodology represents significant progress in fortifying individuals' privacy while simultaneously upholding the effectiveness of facial recognition technologies. The core focus of our project lies in the exploration of homomorphic encryption techniques within the context of the Labeled Faces in the Wild (LFW) face dataset. Here, we perform face recognition utilizing artificial neural networks, encrypting the data using homomorphic means. During the execution of the neural network, we adopt a unique strategy of decrypting the data for each epoch, meticulously safeguarding privacy throughout the entire process.

Moreover, our project delves into the realms of differential privacy and homomorphic encryption, implementing strategic measures to mitigate re-identification risks. The overarching aim is to construct a real-time facial recognition system that not only excels in performance across diverse scenarios but also maintains resilience to potential privacy breaches. Through this initiative, we aspire to contribute substantial insights to the broader discourse surrounding privacy-preserving technology in the field of facial recognition. We underscore the critical importance of thorough evaluation and transparency initiatives, ensuring the responsible and ethical integration of advanced technologies in safeguarding individual privacy.

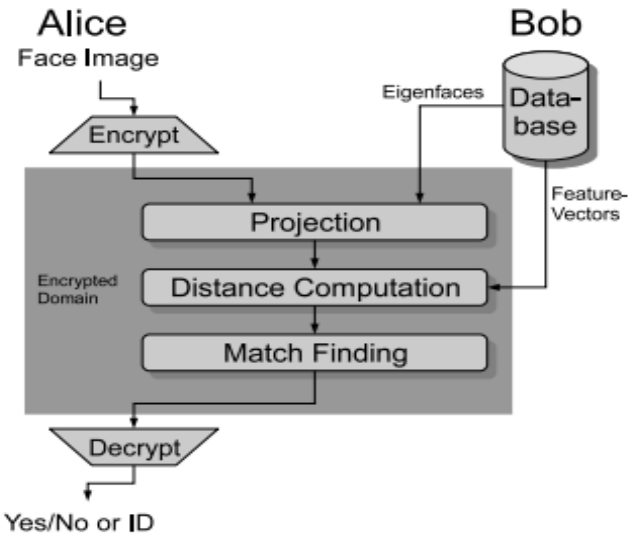

Fig 1 - Basic Encryption Process for Normal Computations (*Google Images*)

2 Related Work

Cryptographic Biometric Template Protection:

Soutar et al. pioneered the integration of biometrics and cryptography with their early system focusing on encrypting fingerprints. This seminal work laid the foundation for enhancing biometric data security. To further this goal, Fully Homomorphic Encryption (FHE) has emerged as a breakthrough solution. Unlike previous methods, FHE doesn't compromise on the utility of biometric templates while ensuring robust security. By maintaining full functionality in the encrypted domain, the FHE solution represents a significant leap forward in cryptographic protection for biometric templates. The innovation holds promise for bolstering the overall security and effectiveness of biometric authentication systems.

Pattern-Recognition-Based Biometric Template Protection:

Pattern recognition-based approaches, as proposed by Davida et al. and Radha et al., offer alternative strategies for securing biometric templates without relying solely on cryptographic methods. Non-invertible transformation functions and cancelable biometrics provide options for protecting biometric signatures, albeit often involving a trade-off between security and matching performance. In contrast, key-binding systems secure templates by binding them to a secret key. Jain et al.'s paper delves into these techniques, offering a comprehensive exploration of the landscape. While these approaches often involve trade-offs, the Fully Homomorphic Encryption (FHE)-based scheme discussed in this context manages to avoid such compromises. It stands out by providing robust template protection without sacrificing matching accuracy, addressing a critical need in the field of biometric template security.

Homomorphic Encryption for Biometrics:

Homomorphic encryption, a powerful tool in privacy-preserving machine learning, has found application in the realm of biometrics. Early attempts using partial homomorphic encryption (PHE) for binarized templates faced challenges such as communication overhead. Subsequent works applied PHE schemes like the Paillier cryptosystem to extend secure matching to fingerprints and binary iris templates. However, limitations persisted, especially concerning real-valued face templates. The introduction of fully homomorphic encryption (FHE) by Gentry et al. opened new possibilities for statistical analysis of encrypted data. This paper proposes a secure face-matching scheme using the Fan-Vercauteren scheme, leveraging batching for efficiency gains. By overcoming computational challenges, this approach significantly enhances the practicality of FHE-based face matching while maintaining data security.

3 METHODS

3.1 DATA COLLECTION

The efficacy of a neural network model in facial recognition is intricately tied to the quality and diversity of the training dataset it is exposed to. The adaptability and generalization capability of the model across a spectrum of facial features, expressions, and environmental conditions hinge on the comprehensiveness of the dataset. Achieving optimal performance necessitates the inclusion of a diverse set of face photos, encompassing variations in ethnicity, age, gender, lighting conditions, and facial expressions. A rich and varied dataset ensures that the model learns robust features and patterns, enhancing its ability to accurately identify faces in real-world scenarios.

One notable example of a dataset used for facial recognition research is the Labeled Faces in the Wild (LFW) dataset. This dataset is curated using the ***fetch_lfw_people*** function, ensuring a minimum of 25 images per person. This deliberate inclusion of multiple images per individual aids in capturing the intra-class variability present in different facial expressions, poses, and lighting conditions associated with a single identity. By printing information about the dataset, including target names (identities or labels) and image shapes, researchers gain valuable insights into the dataset's characteristics. The LFW dataset's diversity makes it a valuable resource for training neural network models, allowing them to better generalize and perform effectively across a wide array of facial recognition tasks, from identity verification to emotion recognition.

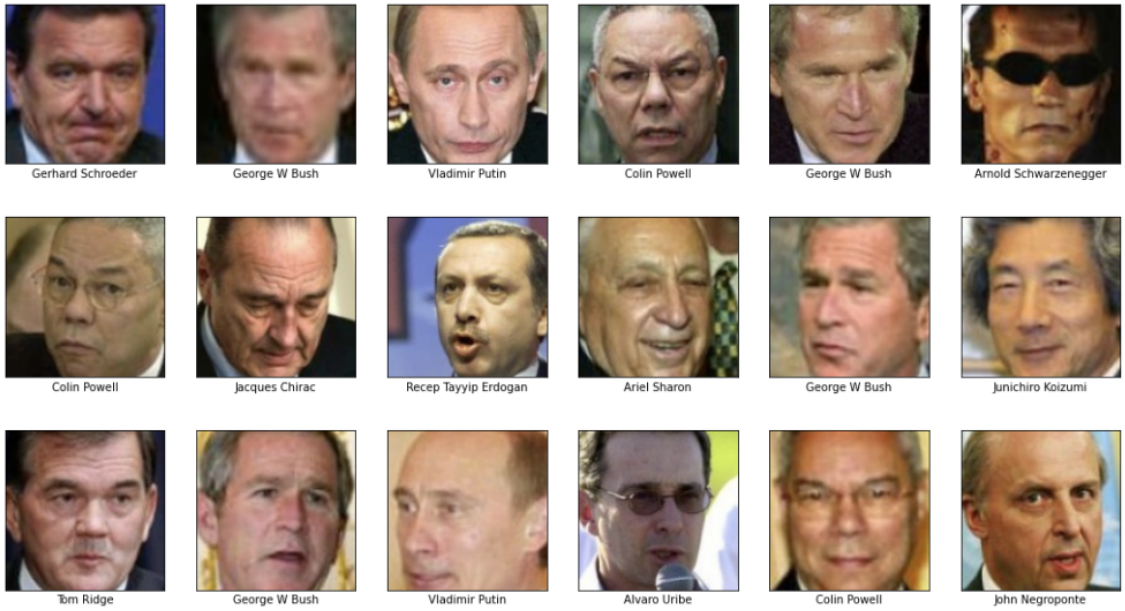

Fig 2 - The people whose gathered images total to greater than 25 in the LFW dataset

3.2 HOMOMORPHIC ENCRYPTION –

Traditional encryption methods aim to safeguard data confidentiality during storage or transmission. Effective encryption ensures that even if unauthorized individuals gain access to encrypted data, they cannot comprehend its contents. In the digital realm, the overarching principle is that encrypted data should appear as random and unintelligible as possible. The more obscured a message is, the less information it reveals. Ideally, perfectly encrypted data divulges no information, and in a robust conventional encryption system, extracting meaningful information from the encrypted data is practically impossible without possessing the decryption key.

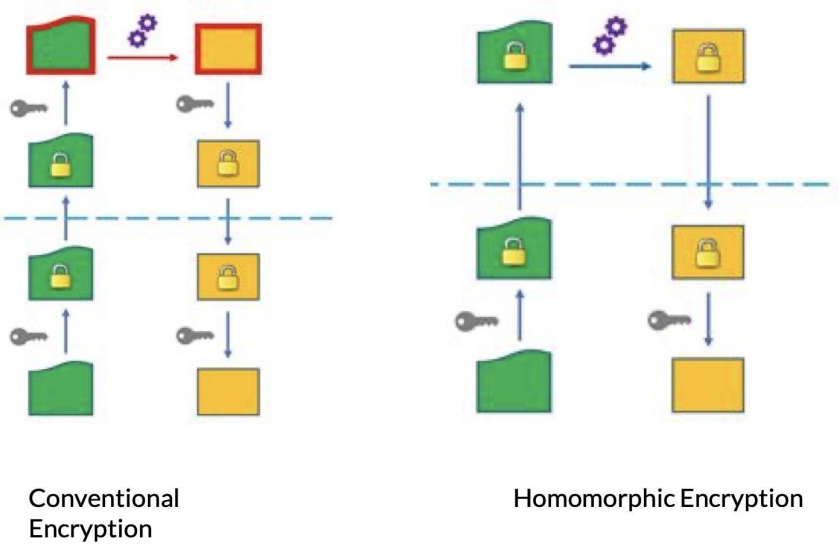

Fig 3 - Difference Between Conventional and Homomorphic Encryption (*Robertdick.org*)

The incorporation of homomorphic encryption presents a novel approach to addressing privacy-preserving and machine learning-related concerns regarding the confidentiality of sensitive data during model training and evaluation. Homomorphic encryption offers a safe framework for information processing while protecting the privacy of the underlying data by allowing computations on encrypted data without decryption.

The practical application of homomorphic encryption in the context of a neural network, specifically designed for facial recognition tasks. The ***TensorSpace (ts)*** library is utilized with the ***Cheon-Kim-Kim-Song (CKKS) scheme***, a homomorphic encryption scheme suitable for real-world applications. The cryptographic context is configured with a ***polynomial modulus degree of 8192*** and coefficient modulus bit sizes carefully chosen as ***[60, 40, 40, 60]***, balancing computational efficiency and security.

The serialized cryptographic context is kept apart as secret and public keys in "***secret.txt***" and "***public.txt***" files, respectively, to aid in future comprehension. For upcoming model deployments, this serialization enables safe encryption key storage and retrieval. To specify the scaling factor for numerical precision during homomorphic operations, the ***global scale parameter*** is set to 2^{40}.

The subsequent sections of the code involve the one-hot encoding of target labels (***y_train, y_val, y_test***) and normalization of input features (***X_train, X_val, X_test***) to a range between 0 and 1. Additionally, the code employs label encoding for the target labels, preparing the data for training the neural network.

The heart of the implementation lies in the encryption of the input features (***enc_X_train, enc_X_val, enc_X_test***) using the CKKS scheme. Each row of the input data is converted into a homomorphic ciphertext vector, ensuring the confidentiality of the facial recognition dataset throughout the entire machine-learning pipeline.

Then, using the ***Keras Sequential API***, a neural network model is built that consists of a softmax output layer for multiclass classification, a dropout layer for regularization, and a dense layer with rectified linear unit (ReLU) activation. With a sparse categorical cross-entropy loss function designed for sparse target labels, the model is compiled using the ***Adam optimizer***.

KEY COMPONENTS OF HOMOMORPHIC ENCRYPTION ----

1. CONTEXT INITIALIZATION:

```
context = ts.context(
    ts.SCHEME_TYPE.CKKS,
    poly_modulus_degree=8192,
    coeff_mod_bit_sizes=[60, 40, 40, 60]
)
```

Initializing a context for homomorphic encryption using the ***CKKS (Cryptographic Key Management Scheme) scheme***. The parameters, such as ***poly_modulus_degree*** and ***coeff_mod_bit_sizes***, define the security and precision of the encryption.

2. GALOIS KEY GENERATION:

```
context.generate_galois_keys()
```

Galois keys are essential for performing certain operations on encrypted data, and here they are generated within the homomorphic encryption context.

3. SERIALIZATION AND STORAGE OF ENCRYPTION CONTEXT FOLLOWED BY DESERIALIZATION

```
secret_context = context.serialize(save_secret_key=True)
public_context = context.serialize()
write_data("secret.txt", secret_context)
write_data("public.txt", public_context)
context = ts.context_from(read_data("secret.txt"))
```

The encryption context is serialized and stored, with the option to save the secret key in the '***secret.txt***' file. The public context can be shared openly, but the secret context should be kept confidential. The encryption context can be deserialized from the stored secret key when needed.

4. MODEL TRAINING & TESTING ON ENCRYPTED DATA

```
history = model.fit(
  np.array([vec.decrypt() for vec in enc_X_train]),
y_train_encoded,
  epochs=50,
  validation_data=(np.array([vec.decrypt() for vec in
enc_X_val]), y_val_encoded),
  shuffle=True
)


test_loss, test_accuracy =
model.evaluate(np.array([vec.decrypt() for vec in
enc_X_test]), y_test_encoded)
print(f'\nTest Accuracy: {test_accuracy}')

predictions = model.predict(X_test)
y_pred = np.argmax(predictions, axis=1)
```

The model is trained using encrypted training data, and decryption is performed during training. The trained model is evaluated on encrypted test data, and the accuracy is printed. This step demonstrates the ability to perform predictions on encrypted data.

3.3 SECURE PROCESSING (IMAGE PROCESSING USING NEURAL NETWORK) -

The neural network model is designed for facial recognition, and it consists of convolutional layers, pooling layers, and fully connected layers.

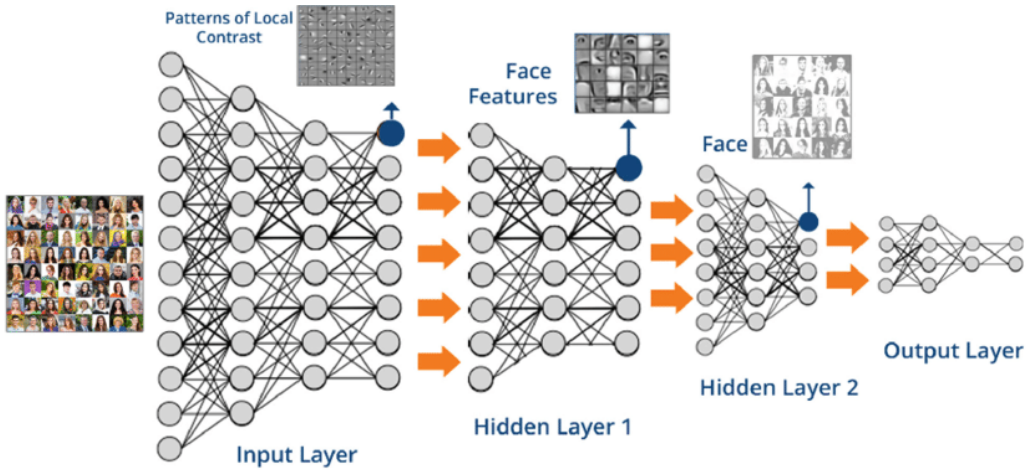

Fig 4 - Neural Network Architecture Simulation (similar to the Network in the Project)

INPUT LAYER-

The input layer in the neural network architecture is implicitly represented by the ***Resizing(224, 224)*** layer. This layer plays a crucial role in preparing the input data for processing by the subsequent layers, particularly the ResNet50 base model.

The ***Resizing(224, 224)*** layer serves as the initial transformation applied to the input images. Its primary purpose is to standardize the dimensions of the input images to ***224x224*** pixels. This standardization is essential because the ResNet50 model, pre-trained on the ImageNet dataset, has been accustomed to processing images of this specific size. By ensuring uniformity in the input dimensions, potential distortions or inconsistencies that might arise from variations in image sizes are mitigated.

Image resizing to 224x224 pixels is a popular practice in convolutional neural networks (CNNs), particularly when using pre-trained models. These models frequently have a set input size, and shrinking the input photos to match this size aids in successfully utilizing the knowledge gathered by the pre-trained model. In this scenario, the scaling layer serves as a preprocessing step, adjusting the dataset to the ResNet50 architecture's requirements.

Moreover, this input layer implicitly defines the number of input channels based on the color format of the images. For instance, if the images are in ***RGB format***, each pixel would have three channels (Red, Green, Blue), and this information is implicitly handled by the resizing layer.

CONVOLUTIONAL LAYERS

The neural network architecture detailed above is carefully constructed to harness the power of transfer learning, utilizing the ***ResNet50*** model pre-trained on the ImageNet dataset. Transfer learning is a technique where a model trained on one task is repurposed for a different, but related, task. In this case, ResNet50, a deep convolutional neural network (CNN), has already learned intricate features from a diverse set of images in ImageNet. By excluding the fully connected layers, which are responsible for the final classification, the base ResNet50 model is transformed into a potent feature extractor.

The decision to make the ResNet50 layer weights non-trainable is critical. This choice ensures that the pre-existing knowledge encoded in the weights, obtained from the massive ImageNet dataset, is preserved. Freezing these weights prevents the risk of overfitting the model to the limited dataset at hand, as well as accelerates the training process. Consequently, the model becomes adept at recognizing general features in images, which is crucial when confronted with a smaller dataset specific to the project.

The subsequent construction of the Sequential model follows a logical sequence. The ***Resizing(224, 224)*** layer is introduced to standardize the input images to dimensions compatible with the expectations of the ResNet50 architecture. This step is critical because it establishes uniformity in the dataset, ensuring that the model interprets each image consistently and preventing any distortions that might hinder performance.

The ***ResNet50*** foundation model is used as a feature extractor within the Sequential framework. The deep architecture of ResNet50 excels at capturing hierarchical information in images, making it appropriate for a wide range of image classification applications. The flattened layer that follows converts the multi-dimensional output of the convolutional layers into a flat array that may be processed by typical densely linked layers.

The subsequent ***Dense layer with 1024 units*** and a ***Rectified Linear Unit (ReLU) activation*** function serves as a feature transformer, enhancing the model's capacity to understand complex patterns in the data. The choice of ReLU as the activation function introduces non-linearity into the model, enabling it to learn and represent intricate relationships within the data.

A softmax activation function is used in the final ***Dense layer***, which has a unit count equal to the specified class count. This layer transforms the raw output of the model into probabilities, allowing it to give a likelihood to each class. This design is ideal for multiclass classification problems, in which the goal is to assign an input image to one of several predefined classes.

The model is then compiled using the ***Adam optimizer***, which adapts the learning rate during training, and categorical cross-entropy loss, an appropriate choice for multiclass classification. Monitoring accuracy as a metric during training provides insights into the model's classification performance.

The training process involves feeding the model with the training data ***(X_train and y_train)*** over ***10 epochs***, with a batch size of 10. The inclusion of validation data ***(X_test and y_test)*** during training enables the assessment of the model's generalization performance on data it has not seen before. The resulting training history (hist) can be analyzed to gauge the model's learning trajectory, identify potential overfitting, and make informed decisions about model adjustments or enhancements.

-

3.5 PRIVACY PRESERVATION —

Homomorphic encryption enables computations on encrypted facial data, ensuring that the raw, identifiable information remains confidential throughout the entire processing pipeline. Traditional

facial recognition systems often involve sending unencrypted biometric data to a central server for processing, posing privacy risks. Homomorphic encryption mitigates this risk by performing computations on encrypted data, even during model training and evaluation.
It demonstrates an implementation of privacy preservation techniques, specifically utilizing the Microsoft Simple Encrypted Arithmetic Library (SEAL) for homomorphic encryption in a neural network context. Privacy preservation is crucial, particularly in scenarios where sensitive data, such as facial recognition information, is involved.

A homomorphic encryption context using the CKKS (Cheon-Kim-Kim-Song) scheme. The parameters, including ***poly_modulus_degree*** and ***coeff_mod_bit_sizes***, determine the security and efficiency of the encryption. This context is then used to generate Galois keys for further cryptographic operations. Additionally, the global scale is set, and the secret and public contexts are serialized and saved into "secret.txt" and "public.txt" files, respectively. The secret context is crucial for decryption.

4 LIMITATIONS

Homomorphic encryption, despite its advantages for privacy, has some downsides that lead to the need for decrypting data before feeding it into a neural network. Understanding these limitations is important for balancing privacy concerns with practical implementation:

COMPUTATIONAL OVERHEAD
: When homomorphic encryption is used instead of traditional computations, the computational overhead is increased. More computing power is required for operations on encrypted data, particularly for intricate tasks like neural network training.

LIMITED HOMOMORPHIC OPERATIONS
: Homomorphic encryption supports basic operations but struggles with more complex ones needed for neural network training. Decrypting data before using it in the network is more practical in dealing with these limitations.

COSTS OF COMMUNICATION AND STORAGE
: Compared to unencrypted data, homomorphically encrypted data requires more resources for transmission and storage. The cost of transmitting and storing encrypted data can be decreased by decrypting the data on the server.

KEY MANAGEMENT CHALLENGES
: Managing and securing encryption and decryption keys becomes more complex with homomorphic encryption. Decrypting at the server simplifies key management since the server holds the necessary decryption keys.

SELECTING A HOMOMORPHIC ENCRYPTION SCHEME
: There are trade-offs between various homomorphic encryption schemes about supported operations, computational efficiency, and security. The code's CKKS scheme has limitations but is appropriate for floating-point operations. Achieving the right balance requires managing these trade-offs.

Considering these limitations, the decision to decrypt data before using it in the neural network in the provided code is based on practical considerations. While homomorphic encryption ensures privacy, the computational overhead and limited supported operations make it more practical to perform the core machine learning computations on decrypted data. Our approach finds a balance between privacy preservation and the computational efficiency required for real-world neural network tasks.

4 REFERENCES

- Secure Face Matching Using Fully Homomorphic Encryption {Computer Vision and Pattern Recognition (cs.CV), Vishnu Naresh Boddeti, 2018}
- On the Application of Homomorphic Encryption to Face Identification {P. Drozdowski; N. Buchmann; C. Rathgeb; M. Margraf; C. Busch, 2019}
- Face Security Authentication System Based on Deep Learning and Homomorphic Encryption {Dechao Sun; Hong Huang; Dongsong Zheng; Haoliang Hu; Chunyue Bi; Renfang Wang, 2021}
- Response Time of Cloud-Based Facial Recognition System Utilizing Homomorphic Encryption {Yoshiaki Narusue, Hiroyuki Morikawa, 2023}
- Biometric template protection using cancelable biometrics and visual cryptography techniques {Harkeerat Kaur, Pritee Khanna, 2015}
- Deep Learning in the Field of Biometric Template Protection: An Overview {Christian Rathgeb, Jascha Kolberg, Andreas Uhl, Christoph Busch, 2023}
- Privacy-Preserving Biometric Matching Using Homomorphic Encryption {Gaëtan Pradel, Chris Mitchell, 2021}
- Homomorphic Encryption Resources - https://www.internetsociety.org/resources/doc/2023/homomorphic-encryption/
- Privacy and Homomorphic Encryption Lectures - https://robertdick.org/iesr/lectures/deshpande-vernekar.pdf
- Homomorphic Face Recognition with Tenseal - https://sefiks.com/2021/12/01/homomorphic-facial-recognition-with-tenseal/
- Homomorphic Image Encryption - https://www.researchgate.net/publication/220050091_Homomorphic_image_encryption
- Assistance for Paraphrasing - GPT and Quillbot
- Assistance for Images - Google Images