Privacy-Preserving Face Recognition Using Noised Eigenvectors

1st Bruce L'Horset
add the department
Paris Institute of Digital Technology
Paris, France
bruce.lhorset@eleve.isep.fr

2nd Charles Mailley
missing
missing
missing
missing

3rd Elodie Chen
missing
missing
missing
missing

4th Sara Ricci
Department of Telecommunication
Brno University of Technology
Brno, Czech Republic
0000-0003-0842-4951

***Abstract*—Face recognition technology is widely implemented in authentication and security systems, yet privacy concerns remain a major challenge. Traditional approaches rely on centralized storage and processing of biometric data, exposing users to the risk of data breaches and identity theft. In this work, we propose a privacy-preserving face recognition framework that integrates k-same pixels, eigenfaces, Principal Component Analysis (PCA), and differential privacy to enhance security while maintaining recognition accuracy. Our approach applies: PCA for dimensional reduction and preserve essential facial features, then differential privacy to mask the biometric and block a possible misuse of them. This approach ensures privacy protection without compromising the information necessary for identity verification. We conduct several evaluations on benchmark datasets to analyze the trade-off between privacy guarantees and reconstruction accuracy. Our results show that integrating differential privacy with PCA-based eigenfaces enhances privacy protection while maintaining high recognition performance. Our experiments show that $\epsilon = 0.68$ achieves the best privacy-utility trade-off in PCA-based facial recognition. Moreover, adding noise to projection vectors rather than eigenfaces improves performance by operating in a lower-dimensional space.**

***Index Terms*—K-Same Pixel, Eigenface, Laplace Noise Addition, Differential Privacy, Facial Recognition, Biometric Authentication.**

I. Introduction

Face recognition [1] is increasingly used in security, authentication, and surveillance applications. Advances in image processing and machine learning have enabled accurate and efficient face recognition systems, from unlocking personal devices to large-scale public monitoring. However, the growing reliance on facial biometrics raises significant privacy concerns, as sensitive data can be misused or exposed in centralized databases.

To address these risks, privacy-preserving techniques can be used to prevent unauthorized access while maintaining system utility. Traditional encryption methods can protect stored data, but often involve high computational costs, making real-time recognition impractical. Alternative approaches such as differential privacy [2] and k-anonymity [3] introduce controlled perturbation or anonymization, ensuring that individual identities remain indistinguishable within datasets while allowing pattern recognition.

A. Contribution

In this work, we propose a privacy-preserving framework that applies: 1) k-same pixel as first technique to smoothen face characteristics, 2) PCA for dimensional reduction and extract key facial features, then 3) introduce controlled Laplacian noise to anonymize identities without compromising recognition.

To facilitate this process, we developed a user-friendly Graphical User Interface (GUI) that allows users to apply different anonymization levels and visualize the privacy-utility trade-off. By evaluating methods through Mean Squared Error (MSE) and Structural Similarity Index Measure (SSIM) metrics, we identify optimal approaches for secure face recognition.

II. Architecture and Analysis Method

The system has been built using the Flask package in Python, enabling deployment as either a website server or a desktop application. Flask provides a straightforward and standardized way to create websites in Python, clearly separating the front-end components (HTML, CSS, and JavaScript) from the back-end logic (Python and SQL). Since this is a testing application, no security features like a login page have been implemented. It is designed as a simple download-and-run application.

The primary goal of this application is to serve as an experimental tool, allowing users to adjust various parameters to generate noised images and attempt to reconstruct the resulting images. Currently, the anonymization of an eigenface can be performed in two ways: either on the eigenface image itself or on its corresponding vector representation. Further analysis will determine which approach is optimal for reconstructing images using machine learning algorithms. The outcome of this analysis will guide the decision on whether to store the noised eigenface images or their PCA vectors in the untrusted database.

The secondary goal is to provide users with optimal parameters for each step of the noised image generation process. To accomplish this, the application includes analyses and graphs that justify the choice of specific parameter values.

The image anonymization process follows the steps illustrated in Figure 1. Starting with a list of front-facing

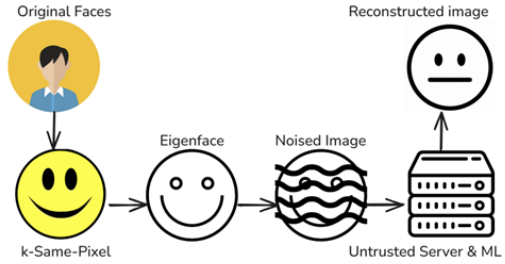

Fig. 1. Anonymization Process Schematic.

photos, the anonymization mechonism applies 1) K-same pixel method, 2) PCA and, finally 3) differential privacy. This produces a list a list of anonymized images that can be stored in an untrusted database. On the other hand, to evaluate the privacy-utility trade-off, the final step involves reconstructing the user's face and comparing the reconstructed image with the original photo provided. The PCA algorithm can produce eigenfaces and eigenvectors ... please specify the prons and cons of both and that we are going to compare them in the experimental results.

III. Back-end and Front-end Architecture

The back-end architecture is designed as an object-oriented application. The main script leverages Flask to operate the local server. The server relies on a Controller object, which orchestrates the various steps of image processing, with each step implemented as a separate class. This approach enables an autonomous workflow for the GUI application while also allowing the creation of multiple analysis objects. These objects can be used to develop and test functions independently, for instance, within Jupyter Notebook or just in a Python console. please describe the steps in the back-end application: take a data set, apply k-pixel, apply PCA, apply DF and then run the ML (specify which ML is used). Once the ML is pass the test phace, it can be used in the front-end application for user authentication... please specify more.

For the front-end, the application runs locally, meaning the client, server, and database are all managed on the user's laptop. However, in a web-based implementation, the database would reside on untrusted third-party servers, as illustrated in Figure 2. For this initial phase, the database is structured as a folder containing noised images for each subject. In future iterations, the images will be migrated to a proper SQL database.

Figure **??** shows the front-end application layout. please add a screenshot (new image) of the app The application comprises three main functional components: adding new users to the database, verifying if a user exists within the database, and managing the database's contents. The new user page enables the creation of new users using the PEEP method, requiring a set of front-facing photos of the same individual. The verification page uses ML algorithms to determine if a user is present in the database based on a single front-facing photo. The database management page allows users to display, remove, or attempt reconstruction of stored images.

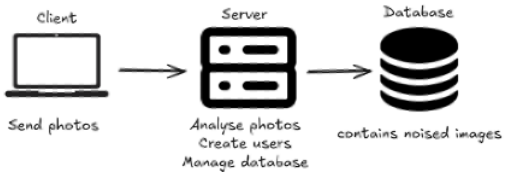

Fig. 2. Front-End Architecture

IV. Experimental Results

This section details the experimental setup, parameter selection, and analysis of the trade-off between privacy and utility. We use MSE and SSIM as our primary metrics. Our experiments were performed on the YaleFace dataset please add here the link to the dataset webpage as a footnote, which contains 11 font-faces phtos of 15 subjects for a total of 165 photos. Our application is not database-dependent and we plan to extend the experimental results considering other databases such as side-faces mixed with fron-faces photos, colored photos, and bigger datasets.

A. PCA Parameter Setup

To determine the optimal value of principal components for the PCA, we conducted experiments using ratios. These ratios represent the proportion of the total number of images per subject that are retained as principal components. We evaluated five specific ratios: 0.4, 0.55, 0.7, 0.85 and 1. For each ratio, we performed PCA on the flattened image data of each subject separately and then reconstructed the images using the corresponding reduced set of principal components. We evaluated the reconstruction quality using MSE and SSIM between the original (resized) images and the reconstructed images (without any added noise). This approach allowed us to isolate the effect of PCA dimensionality reduction, independent of the differential privacy mechanism.

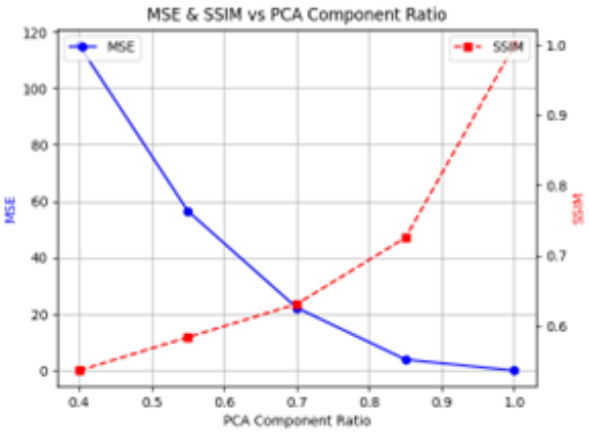

Fig. 3. Impact of Dimensionality Reduction on MSE and SSIM

Figure 3 shows the average MSE for the different ratios and the average SSIM. As expected, increasing the ratio leads to a decrease in MSE and an increase in SSIM, indicating improved reconstruction quality. Specifically, we identify that

- Ratio value = 0.55: At this ratio, we observe a relatively high MSE and a lower SSIM, indicating noticeable loss of information.
- Ratio value = 0.85: This ratio provides a significant improvement. The MSE is considerably lower, and the SSIM is considerably higher.
- Ratio value = 1: While using all components results in the lowest MSE and highest SSIM, the improvement from 0.85 to 1.0 is marginal, demonstrating diminishing returns. Furthermore, retaining all components offers no dimensionality reduction, increasing computational costs and potentially leading to overfitting, where the PCA captures noise and minor, subject-specific variations.

Therefore, we selected a ratio of 0.85 for further analysis. This value strikes a good balance between capturing significant data variations, providing dimensionality reduction for computational efficiency, avoiding potential overfitting, and achieving quality close to the theoretical maximum.

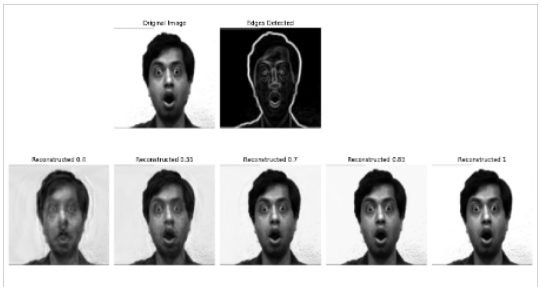

Fig. 4. Example Reconstruction for a Single Subject

B. ϵ Parameter Setup

We evaluated the system's performance under different privacy guarantees by varying the epsilon (ϵ) parameter for the differential privacy method. Specifically, we used epsilon values of 0.5, 1, 2, 4 and 8. These values represent a range from stronger privacy (lower values) to weaker privacy (higher values). To comprehensively assess the interaction between privacy and dimensionality reduction, each epsilon value was tested in combination with each PCA component ratio. The PEEP algorithm was applied to the YaleFace dataset, with Laplace noise added to the projected image data.

Figure 5 and 6 present the MSE and SSIM results, illustrating the relationship between epsilon, PCA component ratio, and reconstruction quality. We observed a consistent trend across all PCA component ratios: increasing epsilon leads to a decrease in MSE and an increase in SSIM. This behavior is fundamental to differential privacy; higher epsilon values allow for less noise to be added, resulting in better image reconstruction but weaker privacy guarantees. The influence of PCA component ratio is also evident - at a fixed epsilon, a higher ratio generally improves reconstruction quality.

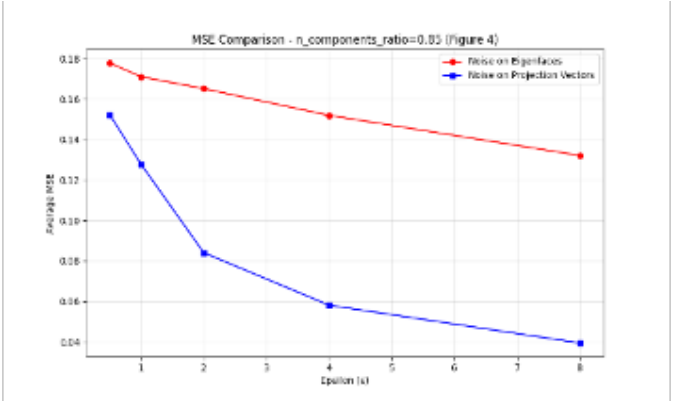

Fig. 5. MSE Comparison for Different Epsilon Values (n_ratio = 0.85)

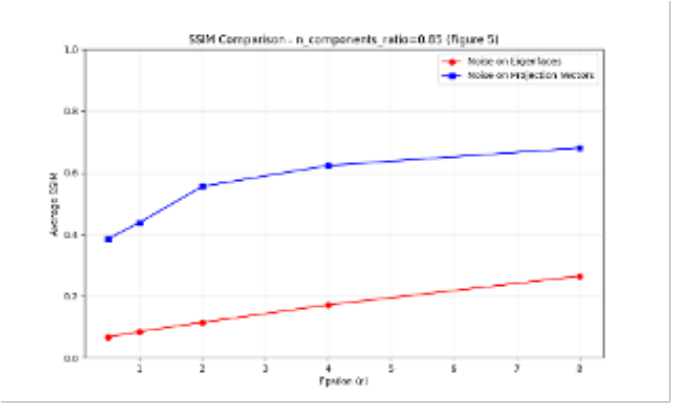

Fig. 6. SSIM Comparison for Different Epsilon Values (n_ratio = 0.85)

C. Eigenfaces vs Eigenvectors Comparison

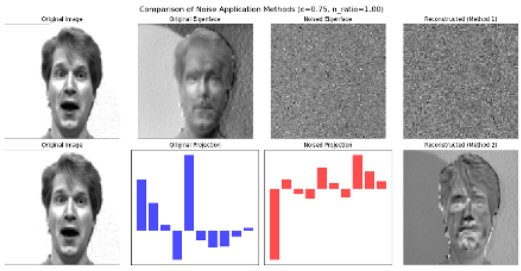

Fig. 7. Comparison of Noise Application Methods ($\epsilon = 0.75$, n_ratio=1)

Figure 7 shows examples of reconstructions for different epsilon values. We compared two approaches for applying differential privacy: adding noise directly to eigenfaces versus adding noise to the projection vectors. Our analysis revealed that the projection vector noise method consistently produces lower MSE values and higher SSIM values across all epsilon settings compared to the eigenface noise method.

This superior performance is largely due to the dimensionality advantage - when applying noise to projection vectors, we work in a lower-dimensional space (typically 8-9 dimensions with our chosen ratio of 0.85) compared to the full eigenface

image space (10,000 dimensions for 100×100 pixel images). This reduced dimensionality means less noise is required to achieve the same privacy guarantees, resulting in better utility. Based on these findings, we selected the projection vector noise method as our preferred approach.

D. Privacy-Utility Trade-off

After selecting the projection vector noise method with n_ratio=0.85, we analyzed the trade-off between privacy protection and recognition utility.

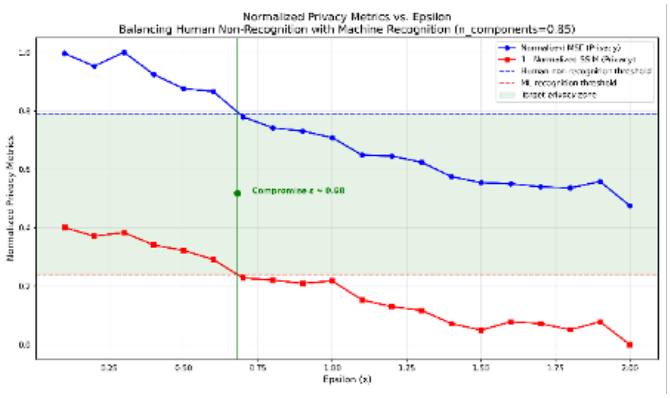

Fig. 8. Privacy-Utility Trade-off: Human vs. Machine Recognition

Fig. 8 illustrates this relationship by plotting normalized privacy metrics against increasing epsilon values. The blue curve represents the normalized MSE (higher values indicate better privacy protection), while the red curve shows 1 - normalized SSIM (also representing privacy protection). The horizontal dashed lines indicate our defined thresholds: the blue line at 0.79 represents the minimum MSE needed to prevent human recognition, while the red line at 0.24 represents the maximum allowed privacy level that still permits machine learning recognition.

These thresholds create a ”target privacy zone” (highlighted in green) where the system achieves both objectives. Our analysis identifies $\epsilon = 0.68$ as the compromise value that best balances these competing requirements. At this epsilon value: 1) the MSE is sufficiently high to distort the facial image enough that humans would have difficulty identifying the subject and 2) the structural similarity remains sufficient for machine learning algorithms to successfully match the face to the correct identity

This value represents the point where the blue curve (MSE) crosses our human non-recognition threshold, providing the minimum level of distortion needed for privacy protection while remaining within the viable range for machine recognition.

V. Conclusion

To conclude, our experiments highlighted the trade-off between reconstruction quality and privacy protection in facial recognition using PCA and a differential noise mechanism. The analysis of principal component ratios showed that a value of 0.85 provided a good balance between dimensionality reduction and reconstruction fidelity while avoiding overfitting. Additionally, the application of noise through the ϵ parameter confirmed that higher values ensure better reconstruction at the cost of weaker privacy. Our results demonstrate that adding noise to projection vectors rather than eigenfaces offers superior performance due to working in a lower-dimensional space.

The most significant finding is the identification of $\epsilon = 0.68$ as the optimal privacy-utility compromise, creating a ”sweet spot” where images are sufficiently distorted to prevent human recognition while maintaining enough structural information for machine-based recognition. This parameter selection ensures that facial recognition remains functional for authorized computational systems while providing meaningful privacy guarantees against human visual identification.

For future work, we plan to explore the k-same pixel approach to further enhance privacy while maintaining recognition performance. Additionally, we will integrate machine learning-based face recognition models for internal testing, analyzing how different parameter variations affect recognition accuracy and privacy guarantees. These next steps aim to refine the privacy-utility trade-off and improve the robustness of privacy-preserving face recognition techniques.

References

[1] Taskiran M, Kahraman N, Erdem CE. Face recognition: Past, present and future (a review). Digital Signal Processing. 2020 Nov 1;106:102809.

[2] Chamikara MA, Bertok P, Khalil I, Liu D, Camtepe S. Privacy preserving face recognition utilizing differential privacy. Computers & Security. 2020 Oct 1;97:101951.

[3] Meden B, Emersic Z, Struc V, Peer P. k-Same-Net: k-Anonymity with generative deep neural networks for face deidentification. Entropy. 2018 Jan 13;20(1):60.