Wrocław University of Science and Technology
Faculty of Information and Communication Technology

Field of study: **Algorithmic Computer Science (INA)**
Speciality: **Algorithmics (ALG)**

MASTER THESIS

Privacy preservation aspects of image modification in facial recognition

Paweł Zalewski

Supervisor
Piotr Syga, PhD

Keywords: facial recognition, embeddings, privacy

WROCŁAW 2024

Abstract

This thesis aims to develop a facial recognition system that enhances security by utilizing embeddings extracted from specifically modified images. The ideal system should perform effectively with modified images while being ineffective with originals. To validate our approach, we employ state-of-the-art neural network models from the face recognition domain, ensuring a comprehensive evaluation across different architectures. Additionally, we investigate various image modification techniques to assess their impact on system performance. By employing data poisoning techniques, particularly the LowKey method, we demonstrate the feasibility of achieving high accuracy with modified images while significantly reducing accuracy with unmodified ones. Our differential privacy analysis highlights the potential of the LowKey modification to enhance users' privacy. These results provide a robust foundation for future research and offer a potential additional layer of security for modern facial recognition systems.

Streszczenie

Celem pracy jest opracowanie systemu rozpoznawania twarzy, który zwiększa bezpieczeństwo użytkowników poprzez wykorzystanie zanurzeń (ang. embeddings) odpowiednio zmodyfikowanych obrazów. Idealny system powinien działać skutecznie z obrazami zmodyfikowanymi, będąc jednocześnie nieskutecznym wobec oryginałów. Aby zweryfikować nasze podejście, wykorzystujemy nowoczesne modele sieci neuronowych z dziedziny rozpoznawania twarzy. Dodatkowo badamy różne techniki modyfikacji obrazów, aby ocenić ich wpływ na wydajność systemu. Stosując zatruwanie danych (ang. data poisoning), w szczególności metodę LowKey, wykazujemy możliwość osiągnięcia wysokiej dokładności systemu dla zmodyfikowanych obrazów, jednocześnie znacząco obniżając dokładność dla obrazów niezmodifikowanych. Nasza analiza prywatności różnicowej (ang. differential privacy) podkreśla potencjał modyfikacji LowKey do zwiększenia prywatności użytkowników systemu. Otrzymane wyniki stanowią solidną podstawę do przyszłych badań i oferują potencjalną dodatkową warstwę bezpieczeństwa dla nowoczesnych systemów rozpoznawania twarzy.

Contents

Introduction

In recent years, the field of machine learning has seen remarkable growth and widespread application across numerous domains. Neural networks have provided state-of-the-art solutions for most computer vision tasks for over a decade, showcasing the versatility and power of these technologies. However, rapid development of deep learning techniques has raised concerns due to the limited interpretability of these models' actions and the enormous amount of data used in their training processes. Since some data might be sensitive, the lack of control over the models can lead to serious privacy violations. This intersection of rapid advancement and potential risk underscores the importance of developing more interpretable and secure artificial intelligence systems. Ensuring the balance between effectiveness and trustworthiness in these systems has become one of the major challenges in modern deep learning. Researchers have recently turned their attention to the concept of embeddings—multidimensional representations of various kinds of input data. The idea behind these vectors is that they should possess useful qualities; for example, embeddings of similar classes should be closer in the embedding space than representations of diverse entities. This concept has become central to numerous advancements in deep learning, as it allows for more meaningful and efficient data representation and processing. This thesis focuses on facial recognition, a subdomain of computer vision that utilizes biometric data (such as face images) for tasks like verification (determining if a person is who they claim to be) and recognition (identifying an individual from a set of possible entities). The idea of embeddings is also vital to this field; vectors are used to efficiently represent a person's face in a space where the similarity between original faces is measured by the distance between their representations. This approach has revolutionized facial recognition systems, making them more accurate and reliable. Nowadays, there is a broad range of resources providing not only neural network architectures and loss functions for effective training of such embedding extractors but also pre-trained models that can be customized even without significant computational power. This accessibility has spurred extensive research on embeddings and led to the conclusion that some features can be reverse-engineered using these high-dimensional vectors. Consequently, in facial recognition systems, the database of embeddings should be treated with special caution, as any potential leakage poses a serious security threat to users. Protecting these embeddings and ensuring users' privacy while maintaining system performance is a crucial aspect of modern facial recognition research.

Problem Statement

The goal of this thesis is to develop a facial recognition system that does not rely on original face embeddings. Instead, we propose modifying the images before feature extraction and retaining their vector representations. The ideal modification should enable the system to perform effectively with similarly modified images while failing to recognize original, unmodified images. In the event of data leakage, embeddings from such a system would provide significantly less information about the users compared to original embeddings. Developing such a system represents a crucial step towards creating better privacy-preserving facial recognition technologies.

Thesis Outline

The first chapter covers the fundamentals, including classification, neural networks, and differential privacy. It provides a literature review on the development and challenges in facial recognition. In the second chapter, the precise approach for achieving the thesis' goal is defined, including a detailed description of the neural network models used, the database, and the image modification techniques. The third chapter presents the experimental results, which include the estimation of the equal error rate threshold, evaluation of system accuracy, differential privacy analysis of the most prominent modification, and sensitivity analysis. The thesis concludes with a summary of the achieved goals and potential future perspectives, emphasizing the significance of our contributions to the field of facial recognition.

Thesis Contribution

We demonstrate that it is possible to use the LowKey modification [6] to create a system with a modified image database that performs well against images with the same modification (74% accuracy) but significantly worse with unmodified images (25% accuracy). We believe this approach can serve as a cornerstone for developing even better techniques focusing primarily on our goal. Furthermore, our differential privacy analysis underscored the potential of the LowKey modification to obscure the presence or absence of an individual in the dataset, thereby enhancing privacy. Specifically, for a 100-class database, the LowKey modification provided 3.1781-differential privacy for GhostFaceNet [2], 3.2581-differential privacy for ArcFace [8], and 3.3322-differential privacy for FaceNet [26]. These results highlight the impact of LowKey in strengthening privacy guarantees. Additionally, we performed a sensitivity analysis on our dataset and concluded that the tested modifications introduce some disturbance in the embedding space, warranting further examination. By integrating these findings, our work lays the groundwork for advancing the development of facial recognition systems that prioritize privacy.

1. Foundations of Facial Recognition

In this chapter, we lay the groundwork for understanding the critical components and challenges associated with facial recognition technology. We begin with a theoretical background, exploring the fundamentals of machine learning, neural networks, and differential privacy. Following this, we review the development of the facial recognition domain, with a particular focus on methods for efficient embedding extraction. We also highlight the challenges faced by facial recognition systems and discuss various strategies for protecting privacy in the context of biometric data.

1.1. Theoretical Background

This section provides a foundational understanding of the key concepts that underpin facial recognition technology, drawing upon important literature, including [23, 14, 10]. We will explore the fundamental principles of machine learning, neural networks, and differential privacy, which form the core of modern facial recognition systems.

1.1.1. Machine Learning

Machine Learning is a data-driven approach to problem-solving that integrates concepts from computer science, probability, and optimization. It involves computational methods aimed at making accurate predictions based on experience, which refers to the past information available to the model [23]. This process of improvement through experience is termed "learning." To formalize this, let us focus on the subdomain of machine learning called supervised learning. Assume that we have random vectors $\mathbf{X} \in \mathbb{R}^n$ and $\mathbf{Y} \in \mathbb{R}^K$ representing data and labels, respectively, along with their joint probability distribution $p_{\mathbf{X},\mathbf{Y}}$, such that $p_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y}) = \Pr(\mathbf{X} = \mathbf{x} \wedge \mathbf{Y} = \mathbf{y})$, where $\Pr$ is the probability measure. Additionally, we define a hypothesis function $h : \mathbb{R}^n \to \mathbb{R}^K$ and a loss function $L : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$, which measures how much the prediction of h differs from the true value $\mathbf{y}$. For a given hypothesis h, we can define the risk as

$$R(h) = \mathbb{E}[L(h(\mathbf{x}), \mathbf{y})] \ . \tag{1.1}$$

The ultimate goal of learning is to find h^* from a fixed family of functions $\mathcal{H}$, such that:

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} R(h) \ . \tag{1.2}$$

In practice, the distribution $p_{\mathbf{X},\mathbf{Y}}$ is usually unknown. Instead, we are given a set of examples $S = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\}$, called the training set. In this situation, we define the empirical risk:

$$\hat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(\mathbf{x}_i), \mathbf{y}_i) \ , \tag{1.3}$$

and look for $\hat{h}$ such that:

$$\hat{h} = \operatorname*{argmin}_{h \in \mathcal{H}} \hat{R}(h) \ . \tag{1.4}$$

Classification

Classification is a fundamental task in machine learning that involves categorizing data into predefined classes. This process typically relies on identifying the underlying distribution of categories based on data features. Common applications include email spam filtering, medical diagnosis, and image recognition. Suppose we have feature vector $\mathbf{x} \in \mathbb{R}^n$ and set of possible classes C, where $|C| = K$ and C_k denotes the k-th class ($k \in \{1, \ldots, K\}$). A basic classification technique, called the Naive Bayes classifier, can be mathematically expressed as:

$$\Pr(C_k \mid \mathbf{x}) = \frac{\Pr(\mathbf{x} \mid C_k) \Pr(C_k)}{\Pr(\mathbf{x})} = \frac{\Pr(C_k)}{\Pr(\mathbf{x})} \prod_{i=1}^{n} \Pr(x_i \mid C_k) \ . \tag{1.5}$$

The crucial assumption is that the features are independent. Despite its simplicity and this assumption, it performs remarkably well even in tasks where the independence assumption is not fulfilled (e.g., spam filtering). A commonly considered classification problem is binary classification, where $K = 2$ and the possible classes are 0 (failure) and 1 (success). The essential method for binary classification is logistic regression, which uses the sigmoid function to estimate the probability of success:

$$\Pr(1 \mid \mathbf{x}; \boldsymbol{\theta}) = \sigma(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + \mathrm{e}^{-\boldsymbol{\theta}^T \mathbf{x}}} \ , \tag{1.6}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n) \in \mathbb{R}^n$ represents the coefficient vector, and σ denotes the logistic sigmoid function. For handling multi-class classification challenges, logistic regression can be extended by running K independent binary logistic regression models, resulting in:

$$\begin{aligned} \Pr(C_K \mid \mathbf{x}; \boldsymbol{\Theta}) &= \frac{1}{1 + \sum\limits_{j=1}^{K-1} \mathrm{e}^{\boldsymbol{\Theta}_j^T \mathbf{x}}} \ , \\ \Pr(C_i \mid \mathbf{x}; \boldsymbol{\Theta}) &= \frac{\mathrm{e}^{\boldsymbol{\Theta}_i^T \mathbf{x}}}{1 + \sum\limits_{j=1}^{K-1} \mathrm{e}^{\boldsymbol{\Theta}_j^T \mathbf{x}}} \ , \end{aligned} \tag{1.7}$$

where $i \in \{1, \dots, K-1\}$ and $\boldsymbol{\Theta} \in \mathbb{R}^{K \times n}$ is the coefficient matrix. Another method for dealing with multiple classes is to use a softmax function, which will be discussed in the section about neural networks. Having basic methods for approximating the underlying distribution of the categories based on data features, it is crucial to evaluate the quality of these predictions. To measure the difference between two probability distributions $\mathbf{p} = (p_1, \dots, p_m)$ and $\mathbf{q} = (q_1, \dots, q_m)$, the Kullback-Leibler (KL) divergence is often considered:

$$D_{\mathrm{KL}}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{m} p_i \log \frac{p_i}{q_i} \ , \tag{1.8}$$

which quantifies the expected information loss when one distribution approximates another. This formula can be expressed using entropy ($H(\mathbf{p})$) and cross-entropy ($H(\mathbf{p}, \mathbf{q})$):

$$H(\mathbf{p}) = \sum_{i=1}^{m} p_i \log \frac{1}{p_i} \ , \tag{1.9}$$

$$H(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{m} p_i \log \frac{1}{q_i} \ , \tag{1.10}$$

$$D_{\mathrm{KL}}(\mathbf{p}, \mathbf{q}) = H(\mathbf{p}, \mathbf{q}) - H(\mathbf{p}) \ . \tag{1.11}$$

Since the goal is to minimize the distance between distributions (thus the KL divergence), but there is no interest in the entropy of those distributions, the usual choice for the loss function in classification problems is cross-entropy.

Gradient Descent

To optimize the model parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ in logistic regression, gradient descent is typically used. Gradient descent is a first-order iterative optimization algorithm for minimizing a function. The parameters are updated in the direction of the negative gradient of the loss function. Let us assume that $\mathbf{x} \in \mathbb{R}^n$ belongs to class C_k and the function $L : C \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is used for evaluating the quality of predictions. The update rule for $\boldsymbol{\theta}$ is given by:

$$\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} L(C_k, \mathbf{x}, \boldsymbol{\theta}) \ , \tag{1.12}$$

where α is the learning rate determining the size of the optimization step. This iterative adjustment of parameters via gradient descent is essential for training logistic regression models, allowing them to efficiently learn from training data and perform robustly in classification tasks.

1.1.2. Neural Networks

In recent years, the focal point of machine learning has increasingly shifted towards a subdomain known as deep learning. This field leverages neural networks to achieve state-of-the-art performance in areas that have challenged traditional algorithms, notably computer vision and natural language processing. Let us take the single pair $(\mathbf{x}, \mathbf{y})$ from the set of observation S. Like before $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^K$. The perceptron is the foundational concept of neural networks—a simple yet powerful structure. This artificial neuron receives multiple input signals, processes them, and produces an output signal. It can be conceptualized as a mechanism for learning a threshold function that maps an input vector $\mathbf{x} \in \mathbb{R}^n$ to an output z:

$$z = h\left(\sum_{i=1}^{m} w_i x_i + b\right) = h(\mathbf{w}^T\mathbf{x} + b) \;, \tag{1.13}$$

where $h : \mathbb{R} \to \mathbb{R}$ represents a non-linear activation function, $\mathbf{w} \in \mathbb{R}^n$ is a vector of weights assigned to the inputs, and $b \in \mathbb{R}$ is the bias. The activation function h must be non-linear to allow the network to learn complex patterns; otherwise, the entire network would be reduced to a linear model, as the composition of linear functions is linear. Neural networks are composed of layers of artificial neurons. The architecture typically includes an input layer, several hidden layers, and an output layer. Each layer's output serves as the input for the next layer, creating a deep network capable of learning intricate patterns and relationships within the data. The process of training neural networks is facilitated by the backpropagation algorithm, which applies the chain rule to compute the gradient of the loss function with respect to each weight in the network. Given a network with L layers, let us define $\mathbf{W}^{(l)}$, $\mathbf{b}^{(l)}$, $h^{(l)}$, $\mathbf{z}^{(l)}$ respectively as weight matrix, bias vector, activation function and the output value of the l-th layer. With $\mathbf{z}^{(0)} = \mathbf{x}$, for $l \in \{1, ..., L\}$ we have:

$$\mathbf{z}^{(l)} = h^{(l)}(\mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}) \;. \tag{1.14}$$

We need a loss function $L : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}$ to measure the difference between the network's output $\mathbf{z}^{(L)}$ and the true label $\mathbf{y}$. To minimize this loss, backpropagation calculates the gradients of the loss function with respect to the weights and biases by applying the chain rule, which allows us to efficiently compute the derivatives through each layer of the network. The gradients are then used to update the parameters through gradient descent, completing the training process. The weight update rule is given by:

$$\mathbf{W}^{(l)} = \mathbf{W}^{(l)} - \alpha \nabla_{\mathbf{W}^{(l)}} L(\mathbf{z}^{(L)}, \mathbf{y}) \;, \tag{1.15}$$

where α is the learning rate. Common activation functions include the sigmoid function, which maps inputs to a range between 0 and 1:

$$\sigma(x_i) = \frac{1}{1 + \mathrm{e}^{-x_i}} \;. \tag{1.16}$$

However, the sigmoid function is prone to the vanishing gradient problem, where gradients become very small during learning, slowing down the training process and potentially leading to poor performance in deep networks. To mitigate this issue, the ReLU (Rectified Linear Unit) is often used, defined as:

$$\mathrm{ReLU}(x_i) = \max(0, x_i) \; . \tag{1.17}$$

The ReLU function helps to address the vanishing gradient problem by allowing gradients to flow more effectively through the network. Nevertheless, ReLU can suffer from the "dying ReLU" problem, where neurons can become inactive and always output zero. To alleviate this, variations such as $\mathrm{Leaky\ ReLU}$ and PReLU (parametric ReLU) are used. $\mathrm{Leaky\ ReLU}$ is defined as:

$$\mathrm{Leaky\ ReLU}(x_i) = \begin{cases} x_i & \text{if } x_i > 0 \; , \\ \alpha x_i & \text{if } x_i \leq 0 \; , \end{cases} \tag{1.18}$$

where α is a small constant. PReLU extends this idea by allowing α to be a trainable parameter. These modifications help to ensure that the neurons continue to learn even when inputs are negative. Finally, the softmax function is often used in the output layer of multi-class classification problems to convert logits into probabilities:

$$\mathrm{softmax}(\mathbf{x}, i) = \frac{\mathrm{e}^{x_i}}{\sum\limits_{j=1}^{n} \mathrm{e}^{x_j}} \; . \tag{1.19}$$

Convolutions

Convolutional neural networks are a special kind of neural networks designed to process data with a known grid-like topology [14]. The most popular examples are 1-D time series or 2-D images (grids of pixels). The core idea behind them is convolution, which is a mathematical operation on two functions of a real-valued argument. However, since in most use cases (including this work), the data considered is discrete, we will focus on discrete convolution. For functions $f_1, g_1 : \mathbb{Z} \to \mathbb{R}$, the discrete convolution can be defined as:

$$(f_1 * g_1)(n) = \sum_{m=-\infty}^{\infty} f_1(m) g_1(n-m) \; . \tag{1.20}$$

Similarly, for $f_2, g_2 : \mathbb{Z}^2 \to \mathbb{R}$, we have:

$$(f_2 * g_2)(n, m) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f_2(i, j) g_2(n-i, m-j) \; . \tag{1.21}$$

In the general case for a group $(G, \oplus)$, where G is discrete, and functions $f, g : G \to \mathbb{R}$, the discrete convolution at a point $a \in G$ is defined by:

$$(f * g)(a) = \sum_{b \in G} f(b) g(a \oplus \hat{b}) \ , \tag{1.22}$$

with $\hat{b}$ such that $b \oplus \hat{b} = 1$, where 1 denotes the identity element in $(G, \oplus)$. A convolutional layer in a neural network consists of several kernels (filters) that convolve with the input data. Each kernel detects different features in the input, and the resulting feature maps are stacked along the depth dimension. This enables the network to learn multiple representations of the data simultaneously. Moreover, the same kernel is used across the entire input, significantly reducing the number of parameters compared to fully connected layers. Convolutions leverage local connectivity by restricting the receptive field of the kernels to small regions of the input. This allows the model to learn spatial hierarchies of features, capturing local patterns in early layers and more abstract representations in deeper layers. Additionally, this means that the learned features are often robust to changes in the position of objects within the input image.

Transfer Learning and Embeddings

The process of training large neural networks, despite optimization methods, remains time and resource-intensive, making it infeasible without substantial computational power. Fortunately, learned features often generalize beyond the training set, allowing for reuse in related tasks. In the field of computer vision, which is the primary focus of this work, classification networks such as VGG [29], ResNet [17], and Inception [30] have been trained on extensive image datasets with thousands of categories. It has been shown that the initial layers of these networks extract universal features (such as shapes and edges), which are combined into more complex structures in the deeper layers. This observation is crucial for a technique called transfer learning, where a model pretrained on a large dataset is repurposed for a different specific task, typically by replacing the final classification layer to match the target dataset. Transfer learning is often combined with fine-tuning, which involves training the final few layers of the network to adapt the model to the new task. The architectures discussed in this work do not contain a classification layer at the end; instead, they focus on creating an embedding of the input. An embedding is a vector $\mathbf{v} \in \mathbb{R}^d$ constructed from an input $\mathbf{x} \in \mathbb{R}^n$ by using a neural network $\Phi : \mathbb{R}^n \to \mathbb{R}^d$, called the feature extractor:

$$\mathbf{v} = \Phi(\mathbf{x}) \ . \tag{1.23}$$

It is often used as a more convenient representation of data in domains like face recognition (to represent faces) and natural language processing (to represent words). Ideally, such vectors should possess intra-class similarity, meaning that embeddings from the same class (or similar classes) should be close to each other in the embedding space. Conversely, different entities should have distant vector representations, which is known as inter-class discrepancy. Methods for achieving these properties will be explored in the next chapter.

1.1.3. Differential Privacy

Differential privacy (DP) is a robust, meaningful, and mathematically rigorous definition of privacy, with algorithms designed to satisfy this definition. It can be viewed as a framework that ensures the inclusion or exclusion of any single individual's data does not significantly affect the outcome of the analysis. The concept was formalized by Cynthia Dwork and colleagues, providing a foundational framework for privacy-preserving data analysis [10]. Before defining differential privacy, few concepts need to be introduced. Given a discrete set B, the probability simplex over B, denoted $\Delta(B)$, is defined as:

$$\Delta(B) = \{x \in \mathbb{R}^{|B|} : \forall_i \, x_i \geq 0 \wedge \sum_{i=1}^{|B|} x_i = 1\} \, . \tag{1.24}$$

A randomized algorithm $\mathcal{M}$ with domain A and discrete range B is associated with a mapping $M : A \to \Delta(B)$. On input $a \in A$, the algorithm $\mathcal{M}$ outputs $\mathcal{M}(A) = b$ with probability $(M(a))_b$ for each $b \in B$. Databases are viewed as collections of records from a universe $\mathcal{D}$. It is often convenient to represent databases by their histograms: $\mathbf{d} \in \mathbb{N}^{|\mathcal{D}|}$, where each entry d_i represents the number of elements in the database of type $i \in \mathcal{D}$. In this representation, the ℓ_1 distance serves as a measure of difference between histograms d and d':

$$\|d - d'\|_1 = \sum_{i \in \mathcal{D}} |d_i - d'_i| \, . \tag{1.25}$$

A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{D}|}$ is (ϵ, δ)-differentially private if, for all $R \subset \mathrm{Range}(\mathcal{M})$ and for all $d, d' \in \mathbb{N}^{|\mathcal{D}|}$ such that $\|d - d'\|_1 \leq 1$:

$$\Pr[\mathcal{M}(d) \in R] \leq \mathrm{e}^{\epsilon} \Pr[\mathcal{M}(d') \in R] + \delta \, . \tag{1.26}$$

If $\delta = 0$, we say that $\mathcal{M}$ is ϵ-differentially private. Moreover, when analyzing a specific algorithm (or operation) $f : \mathbb{N}^{|\mathcal{D}|} \to \mathbb{R}^k$ on the database, it is useful to introduce the ℓ_1-sensitivity of the function f for database d:

$$\Delta_1(f) = \max_{d' \in \mathbb{N}^{|\mathcal{D}|} : \|d - d'\|_1 = 1} \|f(d) - f(d')\|_1 \, . \tag{1.27}$$

Differential privacy can be achieved through various mechanisms designed to maintain privacy guarantees. The Laplace Mechanism adds noise from the Laplace distribution to the output of a function, scaled according to the function's sensitivity, which is particularly useful for numeric queries. The Gaussian Mechanism, often used for functions with real-valued outputs, adds noise from the Gaussian distribution. The Exponential Mechanism selects an output from a discrete set based on a utility function, with probabilities exponentially related to the utility. Differential privacy has broad applications across various domains, including ensuring privacy in statistical databases or protecting training data in machine learning models.

1.2. Facial Recognition

This section provides a literature review on facial recognition technology. We explore different approaches to verification and recognition, the challenges faced by these systems, and the measures taken to protect privacy.

1.2.1. Verification and Recognition

It is worth mentioning that face detection is an integral part of facial recognition systems, responsible for identifying and locating faces within an image. However, since this work focuses on recognition, face detection will not be covered here. There are two main categories of problems connected with the recognition of a particular individual. The first one, called face verification, answers the question of whether two given images represent the same entity. It is widely used in scenarios like authentication in mobile phones. The second category, on which this work is mostly focused, is face recognition, which aims to deduce which person is present in the given image. Both verification and recognition are nowadays solved with very similar core approaches. The first significant work in the field that approached human-level accuracy was DeepFace by Facebook [31]. It used convolutional neural networks with a classification layer followed by a softmax activation function, trained with the cross-entropy loss. While similar works introduced various architectures and face alignment algorithms, their main drawbacks were scalability (the size of the dense classification layer must match the number of different entities in the training dataset) and generalization to unseen identities. A breakthrough was achieved with FaceNet by Google in 2015 [26]. With a training set of 200M images across over 8M identities, it was infeasible to use a straightforward classification layer. Instead, FaceNet focused on image embeddings, training the model to group embeddings of the same person's images closely in the embedding space using L2 distance. This approach reduces the verification and recognition problems to determining whether embeddings are sufficiently close. To achieve this, FaceNet introduced the Triplet Loss function, ensuring that embeddings of the same class are close and those of different classes are far apart. Given a set of face images $X \subset [0,1]^{m \times n \times 3}$, a set of classes C, a function $h : X \to C$ that returns a label for a given image and a set of triplets:

$$T = \{(\mathbf{I}_a, \mathbf{I}_p, \mathbf{I}_n) \in X^3 : h(\mathbf{I}_a) = h(\mathbf{I}_p) \wedge h(\mathbf{I}_a) \neq h(\mathbf{I}_n)\}\ , \tag{1.28}$$

the desired property can be formulated as:

$$\forall_{(\mathbf{I}_a,\mathbf{I}_p,\mathbf{I}_n)\in T} \quad \|\Phi(\mathbf{I}_a)-\Phi(\mathbf{I}_p)\|_2^2+\alpha<\|\Phi(\mathbf{I}_a)-\Phi(\mathbf{I}_n)\|_2^2\ , \tag{1.29}$$

where α is an enforced margin and Φ is the model used for embedding extraction. Training process with The training process with Triplet Loss is challenging and requires carefully selected triplets (discussed further in Chapter 2). Following FaceNet, researchers continued to focus on improving the discriminative power of embeddings. One significant advancement was the introduction of Center Loss [34], which aimed to enhance the discriminative power of the learned features by simultaneously learning a center for each class and penalizing the distances between the deep features and their corresponding class centers . This approach helped to reduce intra-class variations, making the embeddings more robust and effective. Building on the insights from Center Loss [34] and the challenges of Triplet Loss [18], subsequent research further refined the focus on embedding space. This effort was exemplified by the introduction of models such as SphereFace [21], CosFace [33], and ArcFace [8], named after their respective loss functions. These models used modified versions of the softmax loss with margin penalties to enhance embedding separation. All three of these losses can be expressed in a combined formula:

$$L=-\log\frac{\mathrm{e}^{\|\mathbf{v}\|g(\phi_i)}}{\mathrm{e}^{\|\mathbf{v}\|g(\phi_i)}+\sum\limits_{j=1,j\neq i}^{K}\mathrm{e}^{\|\mathbf{v}\|\cos\phi_j}}\ , \tag{1.30}$$

where $\mathbf{v}=\Phi(\mathbf{x})$ is the embedding of the input x. Derivation of this equation from the softmax-loss (combination of the softmax function and cross-entropy loss) will be discussed in detail in Chapter 2. For now, let us denote that $\boldsymbol{\phi}$ represents the angles between embedding $\mathbf{v}$ and particular rows of the weight matrix $\mathbf{W}$ and $g:[0,\pi]\rightarrow[-1,1]$ such that:

$$g(\phi)=\cos(m_1\phi+m_2)-m_3\ , \tag{1.31}$$

where m_1 is the multiplicative angular margin (SphereFace), m_2 is the additive angular margin (ArcFace), and m_3 is the additive cosine margin (CosFace). Many subsequent approaches focused on improving accuracy on benchmark datasets, such as Labeled Faces in the Wild (LFW), which is a benchmark dataset widely used for evaluating face recognition and verification algorithms. However, recent research has shifted from achieving the highest possible accuracy to reducing the number of parameters in models to decrease their complexity. This shift is driven by the practical need to deploy efficient models on embedded devices, such as smartphones and self-driving cars, where computational and memory resources are limited. Deploying large networks like ResNet-50, which has approximately 25 million parameters, on such devices is impractical due to their significant computational and memory requirements. To address this challenge, researchers have developed several techniques and models to create more efficient neural networks. One notable advancement is the introduction of depthwise separable convolutions, popularized by the Xception model [7]. This technique significantly reduces the number of parameters and computational cost compared to standard convolutions. Building on this concept, the MobileNet architecture [19] utilizes depthwise separable convolutions to create lightweight models specifically designed for mobile and embedded applications. Further improvements in efficiency have been achieved with models like GhostNet [16], which employs ghost modules to generate more feature maps from fewer parameters, enhancing both efficiency and performance. These architectures, with slight modifications, serve as backbones for facial recognition systems, such as MobileFaceNets [5] and GhostFaceNets [2].

1.2.2. Challenges in Facial Recognition Systems

To use biometric systems, particularly facial recognition systems, as alternatives to conventional passwords or as identity management systems, they need to be demonstrated as "trustworthy." This section discusses several key concerns that must be addressed to achieve trustworthiness, including performance, bias and fairness, security, explainability, and privacy, based on the insights from [20].

Performance

The first fundamental condition that a system must fulfill is efficacy, which should be robust to various kinds of noise. Nowadays, state-of-the-art systems have recognition performance that exceeds human abilities. However, they should be tested under challenging conditions to safely rely on them. Most of the available data (e.g., captured from surveillance cameras) does not provide good image quality. Even when the image quality is good, the images often exhibit many variations in pose, illumination, and expression (known as *PIE*). Additionally, another significant challenge is the effect of aging on facial recognition performance. As individuals age, their facial features change, which can lead to a decrease in recognition accuracy over time [3]. Effective systems must account for these temporal changes to maintain performance. Other factors influencing the system's performance include the availability of training data and scalability. When it comes to training data, the problem is more complicated than in other computer vision tasks. Besides needing a large, qualitatively labeled database, there must be a careful selection to ensure that all groups of subjects are similarly represented. Otherwise, underrepresentation may lead to biased performance against certain demographic groups. The issue with scalability is that there are not many very large-scale recognition systems, and the existing ones, such as Aadhaar (India's national ID system) [28], NGI (FBI's Next Generation Identification System) [11], and DHS (Centralized Area Video Surveillance System of the U.S. Department of Homeland Security) [32], are restricted in terms of research accessibility.

Security

In the context of facial recognition systems, security refers to the system's resilience against various types of attacks. The first type of attack is the presentation attack, which involves presenting the system with a face that has been altered with special attributes, such as masks or glasses, to deceive it. Next there are adversarial attacks, which introduce digital perturbations—often visually indistinguishable—to images that can significantly compromise the recognition model. While specialized defense mechanisms exist for both types of attacks [25, 15], their effectiveness is limited due to poor generalization across different attack methods; these defenses typically require training with adversarial samples to recognize them effectively. Additionally, there are template attacks, which target face embedding vectors. Research has indicated that these embeddings may reveal information about an individual's attributes [9], and in some cases, can even be used to reconstruct the face from the database [22]. As a result, it is crucial to encrypt embeddings stored in databases, though this introduces challenges for system usability, such as in performing similarity searches. A promising approach is homomorphic encryption, which allows basic operations, like addition and multiplication, to be conducted within the encrypted domain [13]. However, homomorphic encryption remains computationally intensive and requires further optimization. This work aims to address these security concerns by modifying images so that the embeddings extracted from them are significantly different from those of the original images, yet still retain useful for recognition purposes. This approach seeks to enhance the security and robustness of facial recognition systems against the aforementioned types of attacks.

Explainability and Interpretability

A common criticism of deep learning-based systems is their lack of transparency and interpretability. This concern is particularly relevant in recognition and authentication, where it is crucial to understand the model's decision-making process. In the field of computer vision, many studies focus on visualizing the features that contribute to different model outputs [35]. Additionally, some research aims to understand how specific facial attributes are encoded within deep learning embeddings [9]. By comprehending these encodings, we can enhance the fairness and effectiveness of recognition systems. Improved interpretability also fosters trust and confidence among users, ensuring that the systems are seen as reliable and transparent.

1.2.3. Protecting Privacy

Privacy, in contrast to security, does not imply an attacking scenario. It focuses on individuals' data confidentiality and transparency of usage. One significant privacy concern is image scraping, where images are collected from online sources without consent and used to train facial recognition systems [4]. This practice raises ethical and legal issues, highlighting the need for robust regulations to protect individuals' privacy. Regulations such as the General Data Protection Regulation (GDPR) in Europe and the California Consumer Privacy Act (CCPA) in the United States aim to ensure data protection and give individuals control over their personal information. Additionally, techniques like federated learning are being developed to enhance privacy by allowing models to be trained across multiple decentralized devices without sharing raw data [1]. This approach ensures that sensitive data remains on local devices, thereby reducing the risk of unauthorized access and misuse. Besides the security issues and methods for improved storage of fragile data, there is a broad domain of research focused on methods for protecting users from unauthorized facial recognition systems. In such scenarios, we think of users as attackers on deep learning systems, so techniques like adversarial and presentation attacks may be beneficial for users. By incorporating adversarial perturbations or using presentation attacks, users can protect their privacy and avoid unwanted surveillance. The development of such privacy-preserving methods may be important in the ongoing effort to safeguard personal data in the digital age. Attacks discussed previously, such as presentation attacks and adversarial attacks, are of significant interest when it comes to privacy protection. Presentation attacks involve using special equipment, like masks or photos, to deceive facial recognition systems. However, these attacks are often impractical due to the requirement of external objects. On the other hand, adversarial attacks alter images in specific ways to fool the recognition models. Although powerful, these attacks are typically directed towards specific recognition models, making them generalize poorly across different scenarios and defenses. Another research area is data poisoning, which involves modifying images to degrade the performance of models trained on them. This method inspires works that aim to protect users' images shared online by applying perturbations, known as cloaks, to make them less useful for recognition [27, 6]. While such approaches are promising, their effectiveness is often questioned. Despite recent critiques, such as [24], which argue that these methods may offer a false sense of privacy, the techniques for modifying images they employ could prove beneficial for our work.

2. Facial Recognition with Modified Images

In this chapter, we introduce the concept of a facial recognition system that utilizes modified embeddings. We provide a detailed explanation of the models used for feature extraction, the dataset selected for the experiments and the image modification techniques.

2.1. Approach for Secure Facial Recognition

A Safe way of storing face representations is crucial for modern recognition systems, and there is still much to be done to make this process efficient. We aim to develop a system that operates on modified versions of embeddings. While alterations can be applied at various stages of the facial recognition pipeline, we focus specifically on image modifications in this work. Ideally, our system should achieve satisfactory performance with embeddings from modified images while performing poorly with feature vectors extracted from original images. This ensures that the leakage of vectors reveals significantly less information about the users compared to standard methods. Achieving such property could be a significant step towards developing a facial recognition system resistant to template attacks. Before diving into the detailed description of the feature extraction models, dataset, and methods for image modifications, we need to establish a way to evaluate different methods. The general approach for classifying images in face recognition systems involves extracting embeddings and comparing them with those in a database to find the closest neighbor. However, since the system should prevent access by unauthorized entities, there must be a threshold value $t \in [0, 1]$ for the distance metric, above which a person is classified as unknown. Let $\mathbf{I} \in [0, 1]^{m \times n \times 3}$ be the normalized input image of size $m \times n$, F the recognition system (with F_t denoting the system with a specified threshold value), $V \subset \mathbb{R}^d$ the set of available embeddings, C the set of classes (entities), $h : V \to C$ a function that assigns classes to available embeddings, $\Phi : [0, 1]^{m \times n \times 3} \to \mathbb{R}^d$ the model used for embedding extraction, and $\delta : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ the distance metric. The system's classification process for an input image $\mathbf{I}$ can be expressed as follows:

$$F_t(\mathbf{I}) = \begin{cases} h(\mathbf{v}') & \text{if } \delta(\Phi(\mathbf{I}), \mathbf{v}') \leq t \ , \\ \text{unknown} & \text{if } \delta(\Phi(\mathbf{I}), \mathbf{v}') > t \ , \end{cases} \tag{2.1}$$

where $\mathbf{v}' = \operatorname{argmin}_{\mathbf{v} \in V} \delta(\Phi(\mathbf{I}), \mathbf{v})$.

To choose the proper value for the threshold t, we need to focus on two important metrics: False Acceptance Rate (FAR) and False Rejection Rate (FRR). Let us denote $X_A \subset [0,1]^{m \times n \times 3}$ and $X_U \subset [0,1]^{m \times n \times 3}$ as sets of images of authorized and unauthorized entities respectively. The FAR calculates the probability of a biometric system allowing unauthorized user access and represents the fraction of users from outside the database that are incorrectly accepted:

$$\mathrm{FAR}(F_t) = \frac{|\{\mathbf{I} \in X_U : F_t(\mathbf{I}) \neq \text{unknown}\}|}{|X_U|} \; . \tag{2.2}$$

Conversely, the FRR calculates the probability of a biometric system rejecting authorized user access and represents the fraction of users from the database that are incorrectly rejected:

$$\mathrm{FRR}(F_t) = \frac{|\{\mathbf{I} \in X_A : F_t(\mathbf{I}) = \text{unknown}\}|}{|X_A|} \; . \tag{2.3}$$

It is easy to observe that increasing the threshold will inevitably increase the FRR and decrease the FAR, because it will be more difficult for a system to find a neighbor for the embedding. Similarly, decreasing the threshold will decrease the FRR but increase the FAR. The importance of each metric completely depends on the task and cannot be determined unanimously. A popular choice for biometric systems, which we will use in this work, is finding the threshold $t' \in [0,1]$ for which the value of FAR and FRR is equal:

$$\mathrm{FAR}(F_{t'}) = \mathrm{FRR}(F_{t'}) \; . \tag{2.4}$$

The value of these metrics for the t' is called the Equal Error Rate (EER). As mentioned before, we want to evaluate each system with a database of modified images versus: known and unknown identities with unmodified images, aiming for low accuracy and high EER; and known and unknown identities with images modified in the same way, aiming for high accuracy and low EER. To achieve this, we need to establish the t' for each system and use the accuracy obtained for that t' to compare the systems created with different models and modifications.

2.2. Model Selection

To achieve more meaningful and trustworthy results, we decided not to rely on only one facial recognition model, but to compare three of them: FaceNet, ArcFace and GhostFaceNet. The choice was also affected by the timeline of the facial recognition development, because each one of the chosen architectures had introduced some kind of a breakthrough in the domain. Deep understanding of them can be potentially beneficial in terms of developing cutting-edge solutions not only in facial recognition and verification, but also in broad domain of computer vision.

2.2.1. FaceNet

FaceNet, introduced by Google in 2015 [26], is one of the pioneering and most influential works that brought significant attention to the embedding space in the face recognition field. Let us define a set of face images $X \subset [0,1]^{m \times n \times 3}$, a set of classes C, a function $h : X \to C$ that returns a label for a given image, a model $\Phi : X \to \mathbb{R}^d$ used for generating embeddings, and a set of triplets:

$$T = \{(\mathbf{I}_a, \mathbf{I}_p, \mathbf{I}_n) \in X^3 \mid h(\mathbf{I}_a) = h(\mathbf{I}_p) \wedge h(\mathbf{I}_a) \neq h(\mathbf{I}_n)\} \;, \tag{2.5}$$

where the images in a triplet are often called an anchor ($\mathbf{I}_a$), a positive ($\mathbf{I}_p$), and a negative ($\mathbf{I}_n$). The loss function introduced in FaceNet is called Triplet Loss and is formulated as:

$$L(\Phi, T) = \sum_{(\mathbf{I}_a, \mathbf{I}_p, \mathbf{I}_n) \in T} \left[\|\Phi(\mathbf{I}_a) - \Phi(\mathbf{I}_p)\|_2^2 - \|\Phi(\mathbf{I}_a) - \Phi(\mathbf{I}_n)\|_2^2 + \alpha \right]_+ \;, \tag{2.6}$$

where $[x]_+$ is equivalent to $\max(0, x)$. The goal of the Triplet Loss is to satisfy the following property:

$$\forall_{(\mathbf{I}_a, \mathbf{I}_p, \mathbf{I}_n) \in T} \quad \|\Phi(\mathbf{I}_a) - \Phi(\mathbf{I}_p)\|_2^2 + \alpha < \|\Phi(\mathbf{I}_a) - \Phi(\mathbf{I}_n)\|_2^2 \;. \tag{2.7}$$

Generating all possible triplets results in many that easily fulfill the constraint 2.7, so a special selection process called "triplet mining" was introduced. One training batch consists of a few thousand images, with around 40 faces per identity and some randomly selected negative examples. All anchor-positive pairs in the batch are used for training, but to prevent poor convergence, the negative $\mathbf{I}_n$ for a pair $(\mathbf{I}_a, \mathbf{I}_p)$ must meet the condition:

$$\|\Phi(\mathbf{I}_a) - \Phi(\mathbf{I}_p)\|_2^2 < \|\Phi(\mathbf{I}_a) - \Phi(\mathbf{I}_n)\|_2^2 < \|\Phi(\mathbf{I}_a) - \Phi(\mathbf{I}_p)\|_2^2 + \alpha \;. \tag{2.8}$$

The model Φ used for feature extraction has about 8M parameters and is based on the Inception-ResNet architecture, which combines the strengths of inception modules and residual connections.

2.2.2. ArcFace

ArcFace, introduced by Deng et al. in 2019 [8], is extending the idea of intra-class compactness and inter-class separability introduced in FaceNet. However, instead of using Triplet Loss it introduces the concept known as additive angular margin. Given that Φ returns an embedding $\mathbf{v} \in \mathbb{R}^d$ of image $\mathbf{I} \in X$ from class C_k. The label $\mathbf{y} \in \mathbb{R}^K$ for $\mathbf{I}$ has 1 on k-th position and 0 on the rest (one-hot encoded vector). In the typical classification scenario $\mathbf{v}$ is transformed into logits vector $\mathbf{p} \in \mathbb{R}^K$:

$$\mathbf{p} = \mathbf{v}\mathbf{W} + \mathbf{b} \;, \tag{2.9}$$

where $\mathbf{W} \in \mathbb{R}^{d \times K}$ is the weight matrix and $\mathbf{b}$ the bias vector. Than the softmax function is applied to convert logits $\mathbf{p}$ into probability distribution $\hat{\mathbf{y}}$, which is then compared with the label $\mathbf{y}$ using the cross-entropy loss:

$$\forall_{i \in \{1,\ldots,K\}} \quad \hat{y}_i = \mathrm{softmax}(\mathbf{p}, i) \ ,$$

$$L(\mathbf{y}, \mathbf{p}) = -\sum_{i=1}^{K} y_i \log(\mathrm{softmax}(\mathbf{p}, i)) = -\sum_{i=1}^{K} y_i \log \hat{\mathbf{y}} \ , \tag{2.10}$$

$$y_k = 1 \wedge \forall_{j=1}^{K} y_j = 0 \implies L(\mathbf{y}, \mathbf{p}) = -\log \frac{\mathrm{e}^{p_k}}{\sum\limits_{j=1}^{K} \mathrm{e}^{p_j}} = -\log \frac{\mathrm{e}^{\mathbf{W}_k^T \mathbf{v} + b_k}}{\sum\limits_{j=1}^{K} \mathrm{e}^{\mathbf{W}_j^T \mathbf{v} + b_j}} \ . \tag{2.11}$$

Such L that combines softmax activation with cross-entropy loss is usually mentioned in literature as softmax-loss. The problem is that it does not encourage the intra-class compactness and inter-class separability for the embeddings. The idea of ArcFace is to set $\mathbf{b} = \mathbf{0}$, normalize $\|\mathbf{W}_j\|_2 = 1$ for each $j \in \{1, ..., K\}$, replace $\mathbf{W}_j^T \mathbf{v}$ with $\|\mathbf{W}_j\|_2 \|\mathbf{v}\|_2 \cos \phi_j$ (by definition) and normalize $\|\mathbf{v}\|_2 = s$:

$$L(\mathbf{y}, \mathbf{p}) = -\log \frac{\mathrm{e}^{s \cos \phi_k}}{\mathrm{e}^{s \cos \phi_k} + \sum\limits_{j=1, j \neq k}^{K} \mathrm{e}^{s \cos \phi_j}} \ . \tag{2.12}$$

At this point the optimization process is all about angles ϕ between weight matrix $\mathbf{W}$ and embedding $\mathbf{v}$. To minimize formula 2.12 angle ϕ_k between $\mathbf{v}$ and k-th column of $\mathbf{W}$ must decrease, while angles between $\mathbf{v}$ and other columns of $\mathbf{W}$ should increase. To emphasize this the final formula for ArcFace loss includes additive angular margin m:

$$L_m(\mathbf{y}, \mathbf{p}) = -\log \frac{\mathrm{e}^{s \cos(\phi_k + m)}}{\mathrm{e}^{s \cos(\phi_k + m)} + \sum\limits_{j=1, j \neq k}^{K} \mathrm{e}^{s \cos \phi_j}} \ . \tag{2.13}$$

In ArcFace, the feature extraction model Φ is typically based on the ResNet-50 architecture, with around 25M parameters.

2.2.3. GhostFaceNet

GhostFaceNet, introduced by Alansari et al. in 2023 [2], is a specialized version of network called GhostNet, optimized for face recognition tasks. GhostNet is a lightweight and efficient neural network architecture designed primarily for mobile and embedded devices by Huawei researchers [16]. Its main goal is to reduce computational cost and memory usage while maintaining high accuracy. The core innovation in GhostNet is the Ghost Module, which generates more feature maps from fewer computations using inexpensive operations like linear transformations to create additional ghost feature maps. GhostFaceNet is designed for real-time face recognition on resource-constrained devices, making it suitable for applications in security, authentication, and user identification. It performs competitively against more resource-intensive models when trained and evaluated on various face recognition datasets. An essential component of GhostNet's architecture is depthwise convolution. Depthwise convolution is a type of convolution operation that drastically reduces the computational load by applying a single convolutional filter per input channel, rather than applying a filter across all channels. When combined with pointwise convolutions (1×1 convolutions), depthwise convolutions form a depthwise separable convolution, further optimizing performance by separating spatial and channel-wise operations. The number of parameters of Φ for a version of GhostFaceNet that perform the best on a benchmarks is around 4M.

2.3. Dataset for Evaluation

Since we are using pretrained models for embedding extraction, we do not need a very large database of face images for training. However, it is crucial to have a database with a sufficient number of images per person. Fortunately, there are datasets available that meet this criterion, often containing images of celebrities that we can utilize. Our choice was the FaceScrub dataset, which includes around 530 individuals with approximately 107,818 face images in total. The variant we used has all faces already aligned, as we do not want to involve face detection in this work. We extracted images from FaceScrub to create subsets suitable for evaluation. These subsets consist of two parts: a database with known embeddings and test embeddings used for evaluation. The test embeddings include an equal number of images from people present in the database and individuals who are not, to properly measure the False Acceptance Rate (FAR) and False Rejection Rate (FRR). For most of the experiments, each person in the dataset was represented by 10 images, while each person in the evaluation set was represented by 30 images.

2.4. Image Modifications

This section explores various methods of image modification, ranging from basic techniques to more advanced ones. Their performance in achieving the goals of this thesis is described in the third chapter.

2.4.1. Basic Techniques

Here we introduce methods which typically involve straightforward alterations that can be easily applied without requiring significant computational resources.

Blur

Blurring is a simple yet effective technique for anonymizing images by obscuring facial features. This method reduces the clarity of the image, making it difficult to recognize individuals while preserving the overall structure and context of the image. Gaussian filters are commonly used for this purpose, as they provide a smooth and natural-looking blur. The size of the Gaussian filter can be adjusted to control the level of blurring, with larger filters resulting in more significant blurring effects.

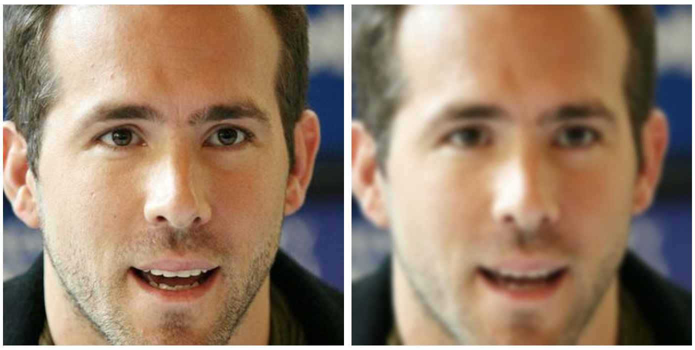

Fig. 2.1. Original image (left) and blurred image (right).

Blocks Permutation

Blocks permutation involves dividing the image into small blocks (in our experiments 40×40 pixels) and then randomly permuting these blocks. This method disrupts the spatial coherence of the image, effectively anonymizing the content by scrambling the facial features across the image. The size of the blocks can be adjusted to control the granularity of the permutation, with smaller blocks resulting in more detailed scrambling. To ensure reproducibility and consistency, a seed value is saved and used for the random permutation process, allowing the same permutation to be applied consistently.

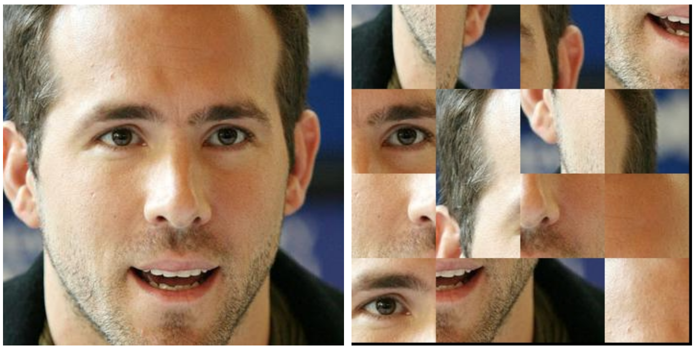

Fig. 2.2. Original image (left) and image with block permutation (right).

2.4.2. Fawkes

Fawkes is a tool designed by Shan et al. in 2020 [27], with the purpose of protecting users' privacy by applying subtle perturbations to images before they are shared online. These perturbations, although subtle, are intended to degrade the performance of unauthorized facial recognition models by disrupting the proper embedding of the person in the image. Despite the popularity of Fawkes, studies like [24] have shown that its assumptions about privacy are often naive, and the protection of users can be illusory. Our interest in this method is due to its similarity to adversarial attacks, which have proven effective over the years in fooling neural networks. If the perturbations are strong enough to make the model mistake the original image for the modified one, but at the same time modified images from the same user share common features, Fawkes may align with the goals of this thesis. Let $\Phi : [0,1]^{m \times n \times 3} \to \mathbb{R}^d$ be the model used for extraction of the embeddings, and $\rho \in [0,1]$ the perceptual perturbation budget. The Structural Similarity Index (SSIM) is a perceptual metric used to measure the similarity between two images $\mathbf{I_1}, \mathbf{I_2} \in [0,1]^{m \times n}$:

$$\mathrm{SSIM}(\mathbf{I_1}, \mathbf{I_2}) = \frac{(2\mu_1\mu_2 + C_1)(2\sigma_{1,2} + C_2)}{(\mu_1^2 + \mu_2^2 + C_1)(\sigma_1^2 + \sigma_2^2 + C_2)} \ , \tag{2.14}$$

where μ_1 and μ_2 are the pixel sample means of $\mathbf{I_1}$ and $\mathbf{I_2}$, σ_1^2 and σ_2^2 are the variances of $\mathbf{I_1}$ and $\mathbf{I_2}$, and $\sigma_{1,2}$ is the covariance of $\mathbf{I_1}$ and $\mathbf{I_2}$. The constants $C_1 = (k_1 L)^2$ and $C_2 = (k_2 L)^2$ are two variables used to stabilize the division with a weak denominator. Usually, L is the dynamic range of the pixel values, and $k_1 = 0.01$ and $k_2 = 0.03$ are default values. The SSIM value ranges from -1 to 1, where 1 indicates perfect structural similarity. To conveniently use the perturbation budget as a threshold for image change, we will use a slight modification called DSSIM (denoted as d):

$$d(\mathbf{I_1}, \mathbf{I_2}) = \mathrm{DSSIM}(\mathbf{I_1}, \mathbf{I_2}) = \frac{1 - \mathrm{SSIM}(\mathbf{I_1}, \mathbf{I_2})}{2} \ . \tag{2.15}$$

The purpose of Fawkes is to generate the modified ("cloaked") version $\mathbf{I}'$ of image $\mathbf{I} \in [0,1]^{m \times n \times 3}$, which should be perceptually similar to the original (controlled by the perceptual budget ρ), but as distant as possible from it in the embedding space. With λ controlling the impact of input perturbation, the optimization process can be formulated as:

$$\max_{\mathbf{I}' \in [0,1]^{m \times n \times 3}} \left[\|\Phi(\mathbf{I}) - \Phi(\mathbf{I}')\|_2 - \lambda \max(d(\mathbf{I}, \mathbf{I}') - \rho, 0) \right] \ , \tag{2.16}$$

and is performed using gradient ascent, which works like gradient descent described in section 1.1.1, just with the opposite sign. The choice of feature extractors is crucial for the performance and effectiveness of privacy-preserving methods like Fawkes. Different architectures and training datasets can significantly impact the robustness and accuracy of the extracted features. In Fawkes, the feature extractors used include InceptionResNet and

DenseNet, trained on datasets like WebFace and VGGFace2. These models are well-known for their capability to generate high-quality embeddings, which are essential for the perturbations to be effective. Fawkes offers different modes of perturbation, allowing users to choose the level of privacy they desire. These modes range from low to high perturbation levels, balancing between image quality and the effectiveness of privacy protection. Figure 2.3 illustrates images with these different modes applied.

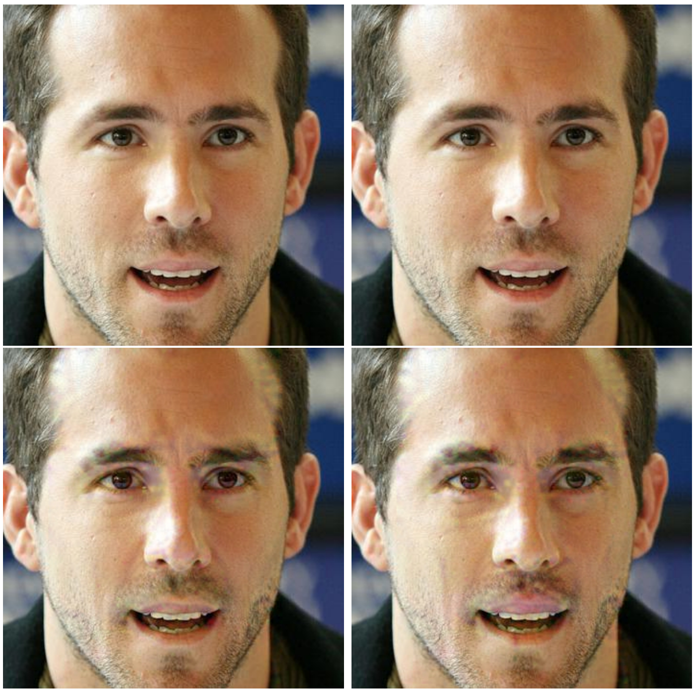

Fig. 2.3. Original image (top left) and images with increasing levels of Fawkes perturbations: low (top right), medium (bottom left), and high (bottom right).

2.4.3. LowKey

LowKey, designed by Cherepanova et al. in 2021 [6], is another data poisoning method designed as a tool for user anonymization. It introduces some new ideas but can be perceived as an enhancement of Fawkes in certain aspects. The first difference in LowKey is the use of the Learned Perceptual Image Patch Similarity (LPIPS) metric instead of DSSIM as a method for measuring perceptual changes. The LPIPS can be formally defined as follows:

$$d(\mathbf{I_1}, \mathbf{I_2}) = \mathrm{LPIPS}(\mathbf{I_1}, \mathbf{I_2}) = \sum_{l=1}^{L} w_l ||\Phi^{(l)}(\mathbf{I_1}) - \Phi^{(l)}(\mathbf{I_2})||_2^2 \,, \tag{2.17}$$

where $\mathbf{I_1}$ and $\mathbf{I_2}$ are the two input images, $\Phi^{(l)}$ returns the feature maps extracted from the l-th layer of a pre-trained neural network with L layers, and $\mathbf{w}$ are the learned weights applied to the distance between the feature maps. This metric leverages deep features from intermediate layers of the network, capturing high-level perceptual information, with higher values indicating greater dissimilarity between images. A significant difference between Fawkes and LowKey lies in the feature extractors. LowKey uses a vector of extractors, denoted as $\mathbf{\Phi}$, and every model in $\mathbf{\Phi}$ is used in the process of generating perturbations. Additionally, every extractor is trained using state-of-the-art face recognition loss functions, such as ArcFace or CosFace. The optimization process in LowKey can be formulated as:

$$\Psi(\Phi, \mathbf{I}, \mathbf{I}') = \frac{\|\Phi(\mathbf{I}) - \Phi(\mathbf{I}')\|_2^2 + \| \Phi(\mathbf{I}) - \Phi(G(\mathbf{I}'))\|_2^2}{\|\Phi(\mathbf{I})\|_2} \ ,$$

$$\max_{\mathbf{I}' \in [0,1]^{m\times n\times 3}} \frac{1}{2|\mathbf{\Phi}|} \sum_{i=1}^{|\mathbf{\Phi}|} [\Psi(\Phi_i, \mathbf{I}, \mathbf{I}') - \lambda\, d(\mathbf{I}, \mathbf{I}')] \ , \tag{2.18}$$

where G is a Gaussian blur, and λ once again controls the impact of perceptual perturbations. Again the selection of feature extractors in LowKey is vital for ensuring its effectiveness in anonymizing user images. LowKey employs a diverse ensemble of feature extractors, including IR-152 and ResNet-152 backbones, which are specifically trained with ArcFace and CosFace heads. This ensemble approach leverages multiple state-of-the-art models to enhance the robustness of the perturbations. Each extractor contributes to the overall effectiveness by capturing different aspects of the facial features, ensuring comprehensive protection against unauthorized recognition. The 2.4 illustrates the impact of LowKey perturbations on an original image.

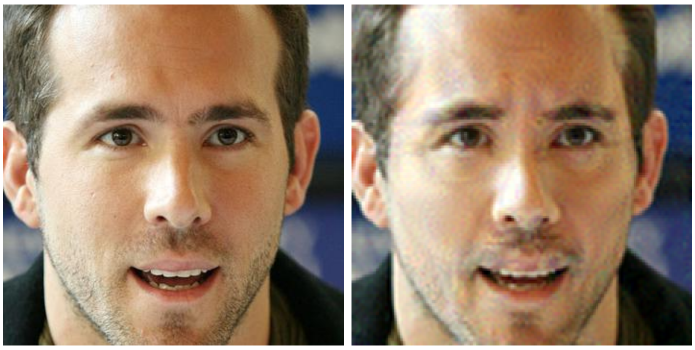

Fig. 2.4. Original image (left) and image with LowKey perturbations (right).

2.4.4. Style Transfer

Popularized by Gatys et al. in 2016 [12], Neural Style Transfer (NST) leverages convolutional neural networks to capture the content of an image $\mathbf{I_c} \in [0, 1]^{m\times n\times 3}$ through high-level feature maps and artistic style of another image $\mathbf{I_s} \in [0, 1]^{m\times n\times 3}$ in order to create $\mathbf{I}' \in [0, 1]^{m\times n\times 3}$ that combines them. The process of NST optimizes a loss function that balances content and style reconstruction:

$$L(\mathbf{I_c}, \mathbf{I_s}, \mathbf{I}') = \alpha L_c(\mathbf{I_c}, \mathbf{I}') + \beta L_s(\mathbf{I_s}, \mathbf{I}') \ , \tag{2.19}$$

where L_c measures the difference in content between the generated and content images, and L_s measures the difference in style between the generated and style images. The content loss L_c is defined as:

$$L_c(\mathbf{I_c}, \mathbf{I'}) = \frac{1}{2} \sum_{i=1}^{N_a} \sum_{j=1}^{M_a} (F(\mathbf{I_c}, a)_{ij} - F(\mathbf{I'}, a)_{ij})^2 \ , \tag{2.20}$$

where F returns the output feature maps of the a-th convolutional layer with N_a feature maps of size M_a. The a is arbitrarily chosen, usually as one of the last layers. This ensures that the generated image retains similar structural elements as the content image. The style loss L_s, where A is the total number of layers, is defined as:

$$L_s(\mathbf{I_s}, \mathbf{I'}) = \sum_{a=0}^{A} w_a E(\mathbf{I_s}, \mathbf{I'}, a) \ , \tag{2.21}$$

where

$$E(\mathbf{I_s}, \mathbf{I'}, a) = \frac{1}{4N_a^2 M_a^2} \sum_{i=1}^{N_a} \sum_{j=1}^{M_a} (G(\mathbf{I_s}, a)_{ij} - G(\mathbf{I'}, a)_{ij})^2 \ , \tag{2.22}$$

and

$$G(\mathbf{I}, a)_{ij} = \sum_{k=1}^{M_a} F(\mathbf{I}, a)_{ik} F(\mathbf{I}, a)_{jk} \ , \tag{2.23}$$

represents the correlations between different filter responses. This encourages the generated image to adopt the textures and color patterns of the style image. In the context of privacy-preserving face recognition, style transfer can be used to alter the appearance of facial images while preserving their identity-related features. By applying the artistic style of a chosen image to a facial image, the resulting stylized image can hinder unauthorized facial recognition models from accurately identifying the person. Figure 2.5 illustrates the impact of style transfer on an image.

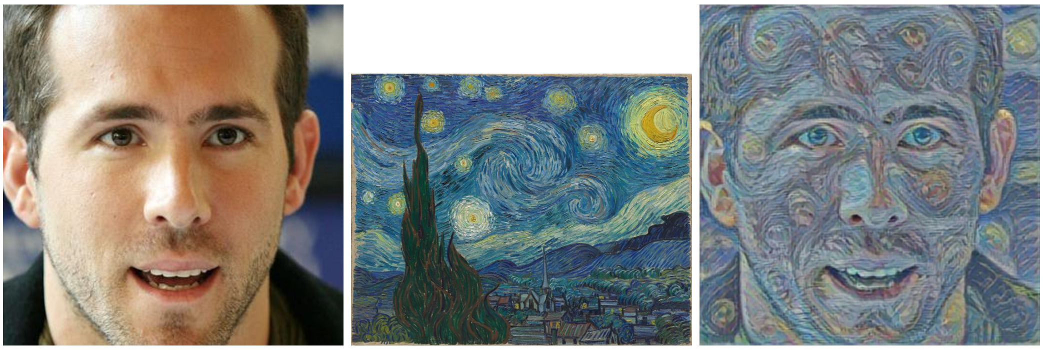

Fig. 2.5. Original image (left), style image (center), and stylized image (right) using Neural Style Transfer.

3. Experimentation and Results

This chapter presents the findings of our experiments. We begin with an estimation of the Equal Error Rate threshold. Following this, we evaluate the level of anonymization provided by each method on a system with an unmodified dataset. The third section, which is the core of this thesis, examines the performance of the system based on embeddings of modified images, evaluated on images modified in the same way and original ones. Finally, we conduct a differential privacy analysis of the most prominent method. These experiments offer a comprehensive understanding of the system's robustness and effectiveness under various scenarios. Detailed experimental results are provided in Appendix A.

3.1. Equal Error Rate Threshold Estimation

To effectively compare different systems, it is necessary to estimate the threshold at which the Equal Error Rate is achieved for each one. Figure 3.1 illustrates an example plot for determining the EER for a system using the ArcFace model. In the subsequent sections, when comparing accuracy, we refer to the accuracy achieved at thresholds where the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) are equal.

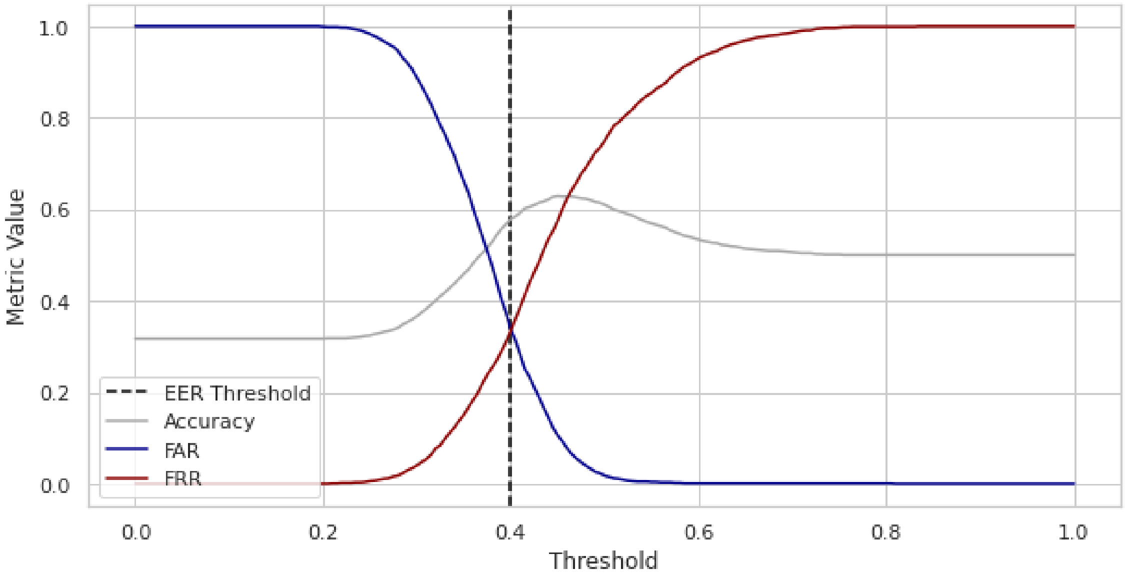

Fig. 3.1. Example estimation of the Equal Error Rate threshold for a system with the ArcFace model architecture, Fawkes as the modification of the database, and unmodified images as the evaluation set.

3.2. Anonymization Evaluation

Before addressing the ultimate goal of this thesis, we want to evaluate the effectiveness of the anonymization provided by the image modifications used. The primary objective of anonymization is to achieve high inaccuracy rates for unauthorized recognition attempts, ensuring that the modified images cannot be easily traced back to the original identities. This effectiveness is measured by the system's inability to recognize modified images when the database consists of unmodified images. In this context, we expect low accuracy in recognizing modified images, which should correspond to a high EER.

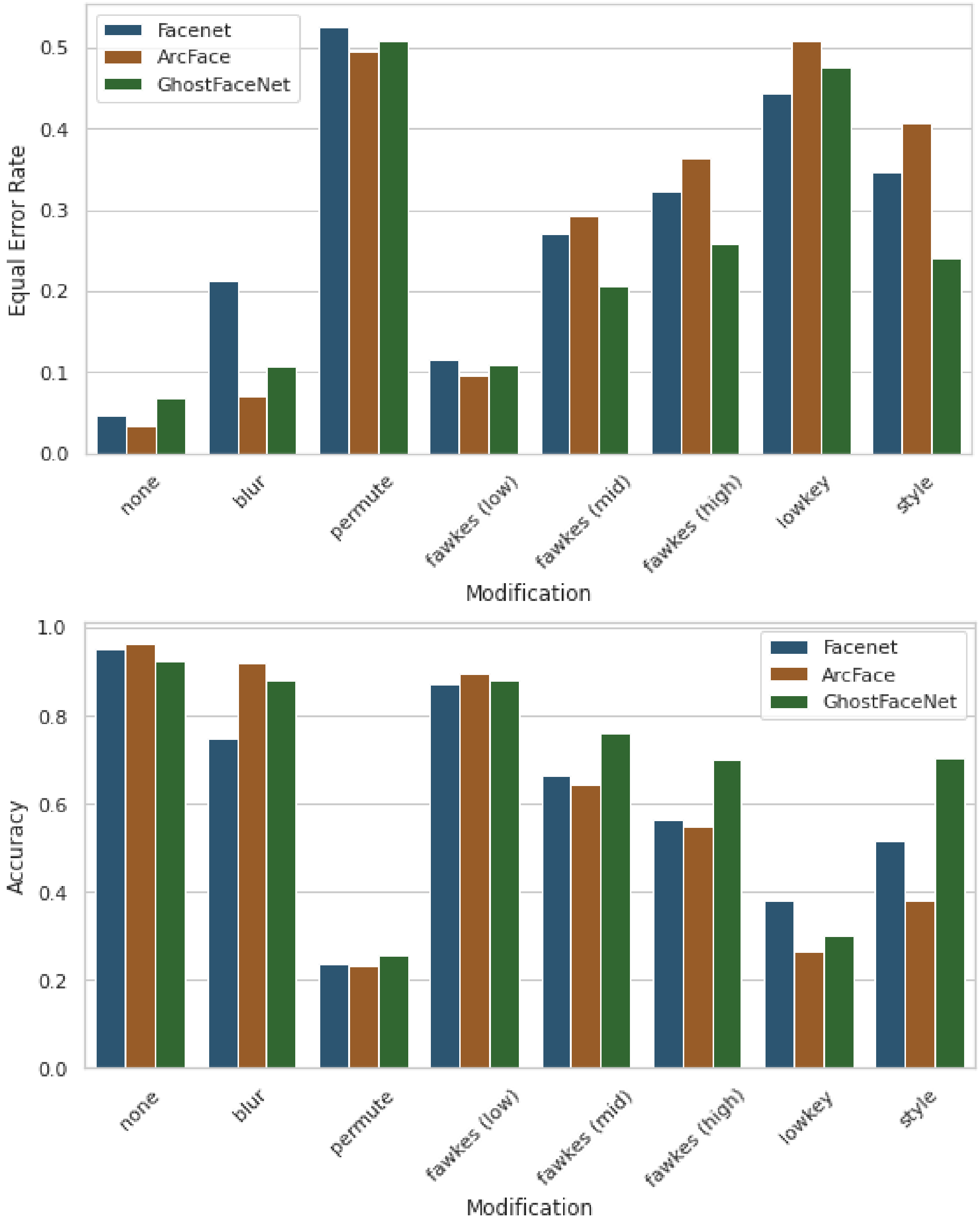

Fig. 3.2. Performance of the system with 50 classes and unmodified database on images with different modifications.

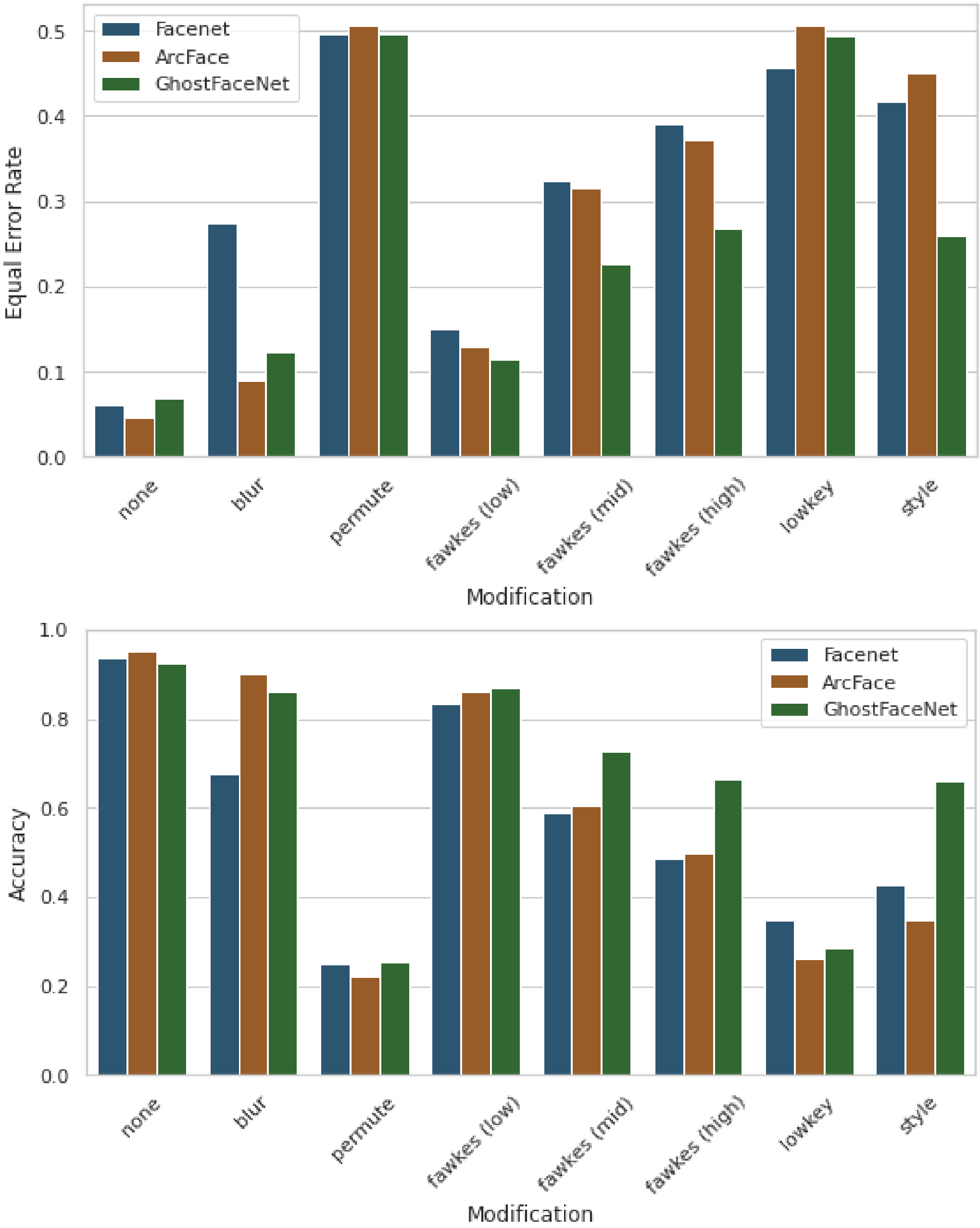

Fig. 3.3. Performance of the system with 100 classes and unmodified database on images with different modifications.

As shown in Figures 3.2 and 3.3, both blur and the weakest Fawkes attack offer almost no anonymization for users. Even with stronger Fawkes attack modes, the system can still achieve an accuracy of over 50% (approximately 66% with the GhostFaceNet model). The situation improves with LowKey, yet the accuracy remains significantly higher than random guessing. The effectiveness of style transfer appears to be highly dependent on the architecture used. The lowest accuracy is observed with block permutations, which is expected due to the substantial impact this modification has on the images. Overall, these findings highlight the varying degrees of effectiveness among different anonymization techniques and underscore the need for further research to develop more robust methods.

3.3. Modified Databases Evaluation

To properly examine the usefulness of a particular modification, we need to evaluate the accuracy of the system using a dataset consisting of images with that modification against both identically modified and unmodified evaluation sets. These experiments are the core of our work, as they directly assess the ability of our system to distinguish between modified and unmodified images, which is crucial for enhancing security. We aim to demonstrate that the system can maintain high performance with modified images while being ineffective with originals. The performance on modified images demonstrates the system's efficacy and ideally should be nearly perfect, indicating that the system can correctly identify users based on their modified images. Conversely, the quality of predictions for unmodified images indicates the risk posed by the leakage of the modified embeddings; in a perfect scenario, this performance should be close to zero. By comparing these two aspects, we can gauge the modification's success in securing the embeddings. Achieving high accuracy with modified images while significantly reducing accuracy with unmodified ones would demonstrate that the modifications effectively protect the original embeddings from unauthorized recognition. This outcome would provide a strong indication that our approach offers an additional layer of security, making the embeddings less useful to potential attackers in the event of a data breach.

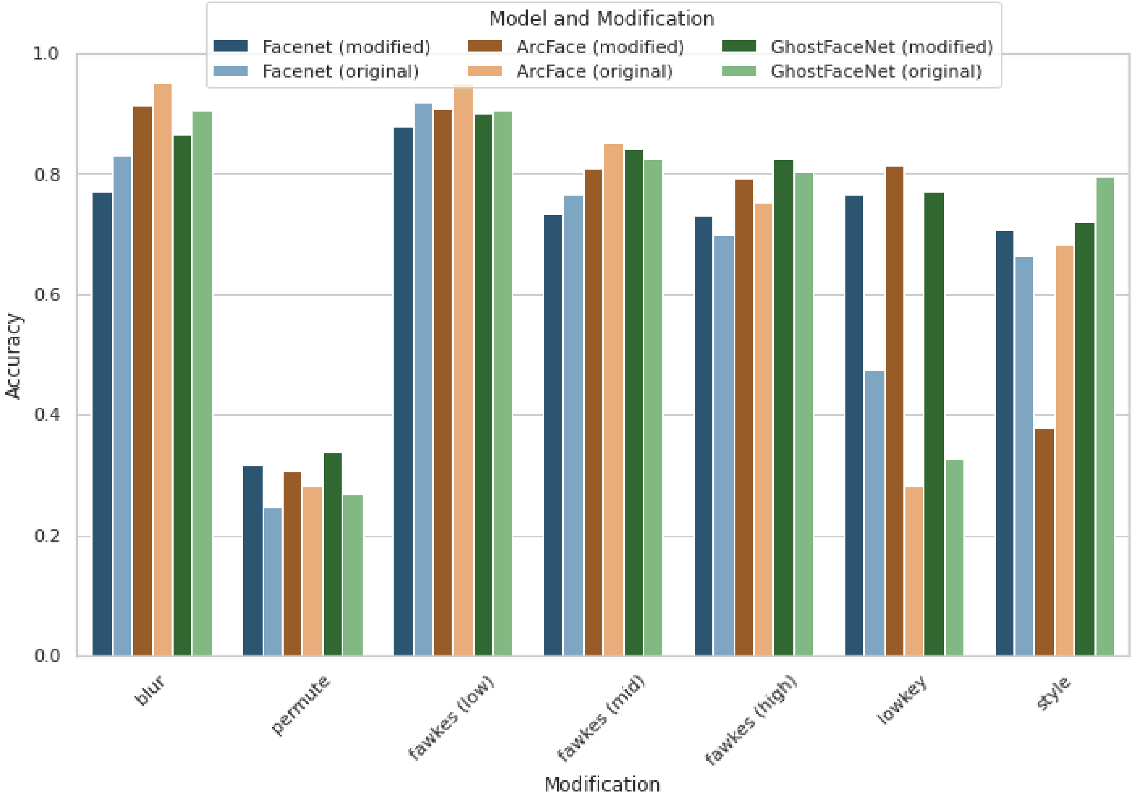

Fig. 3.4. System performance on a modified database of 10 users evaluated using: (1) original images without modifications, and (2) images with the same modifications as the database.

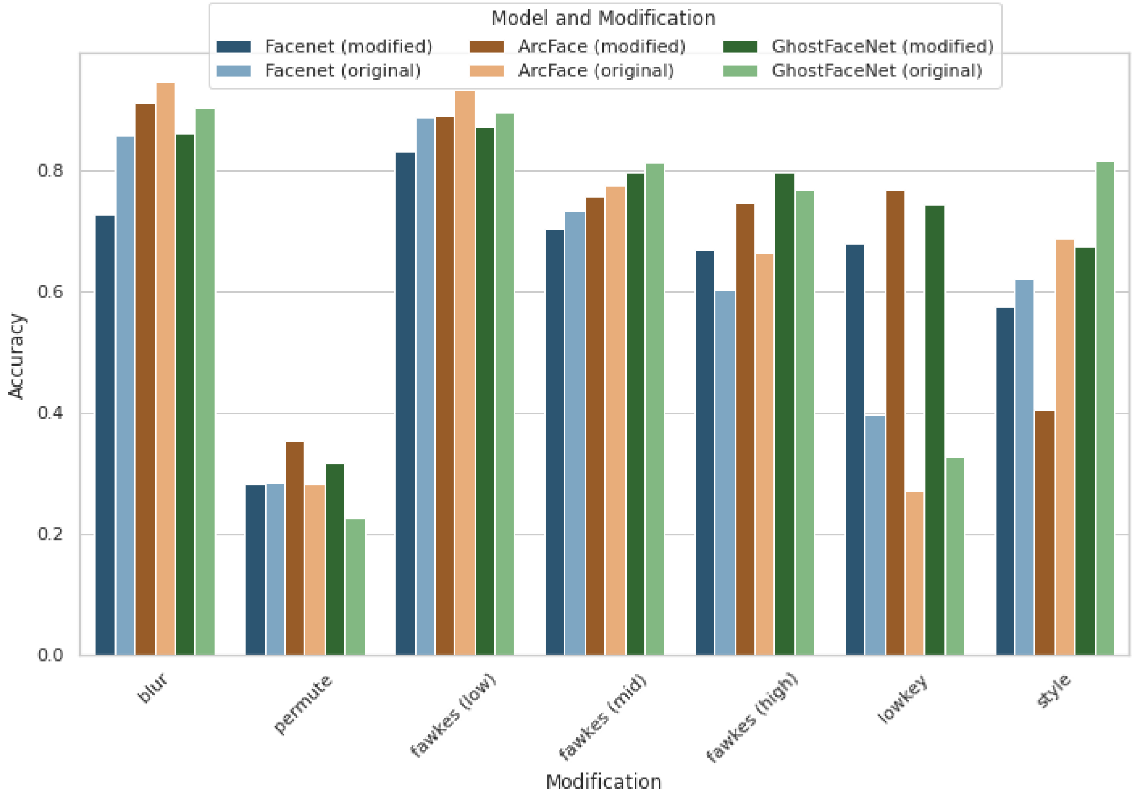

Fig. 3.5. System performance on a modified database of 50 users evaluated using: (1) original images without modifications, and (2) images with the same modifications as the database.

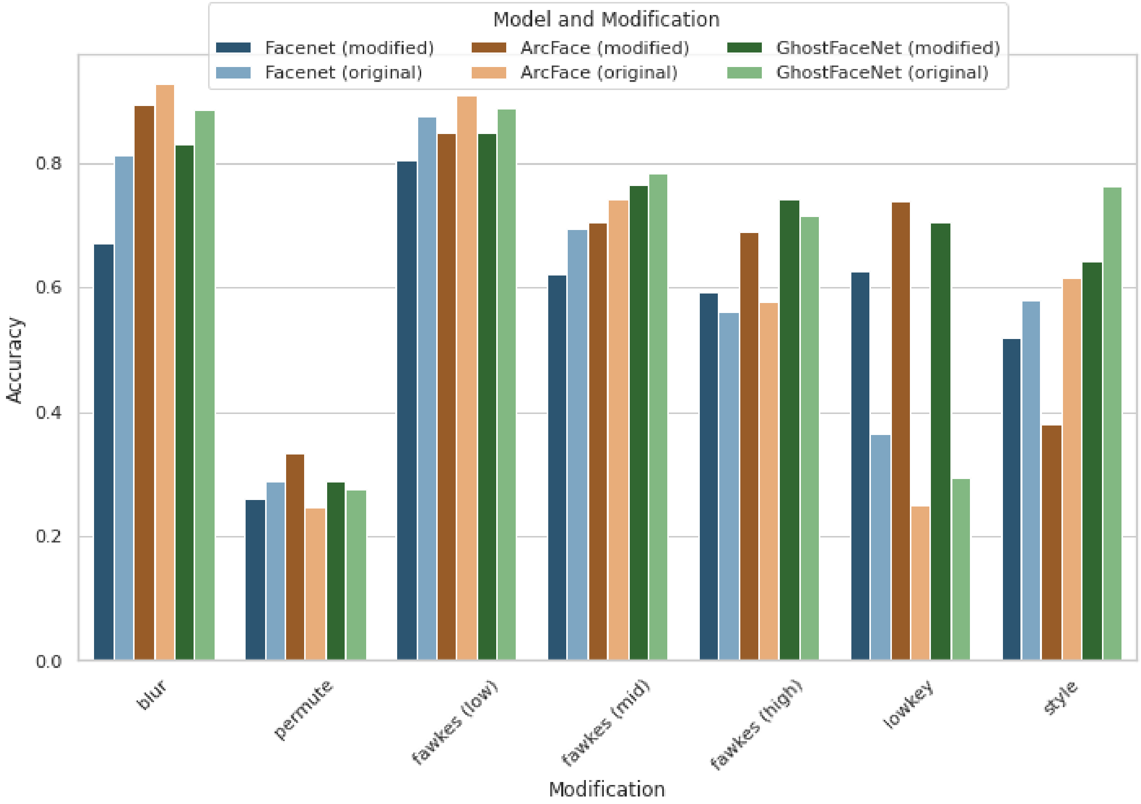

Fig. 3.6. System performance on a modified database of 100 users evaluated using: (1) original images without modifications, and (2) images with the same modifications as the database.

Based on Figures 3.4, 3.5, and 3.6, we can conclude that the system using the LowKey modification shows promising results. Although it does not meet our perfect scenario, there is a significant advantage in accuracy when evaluated with modified images compared to originals. Specifically, the system's accuracy for 100 users drops from 74% to 25% for the ArcFace model, 62% to 36% for FaceNet, and 70% to 25% for GhostFaceNet when the evaluation set is no longer modified. These results highlight the effectiveness of the LowKey modification in providing an additional layer of security by significantly degrading the system's performance on unmodified images. This substantial drop in accuracy indicates that the embeddings generated from modified images are notably different from those generated from original images, thus offering a level of protection against potential misuse in the event of data leakage. Furthermore, the comparative analysis across different neural network models demonstrates the robustness and generalizability of the LowKey modification technique. Each model shows a consistent pattern of performance degradation, reinforcing the reliability of our approach. In contrast, other methods such as blurring and the Fawkes attack, particularly in its weaker forms, do not provide comparable results. These methods fail to significantly reduce the system's accuracy on unmodified images, thereby not offering the same level of protection as the LowKey modification. This disparity underscores the unique effectiveness of LowKey in enhancing the security of facial recognition systems.

3.4. Differential Privacy Analysis

As mentioned in 1.1.3, differential privacy provides a robust framework for quantifying privacy guarantees, ensuring that the inclusion or exclusion of a single data point does not significantly affect the outcome of any analysis. In this section, we analyze the differential privacy properties of the LowKey modification method, specifically evaluating whether it can enhance differential privacy in facial recognition systems. In the domain of facial recognition, the ideal system should accept authorized users and reject unauthorized ones. The impact of including or excluding a specific individual can be evaluated by observing the system's response to that user's face image. The results from previous experiments suggest that a system with a LowKey-modified database may obscure the answer to this question, thereby enhancing privacy. To tailor the analysis to our specific context, we modify the general differential privacy formulas. Instead of considering all possible datasets of users, we focus on a particular dataset D and create a set of neighboring datasets $\mathcal{D}$ by removing one individual from D. Additionally, we concentrate on ϵ-differential privacy ($\delta = 0$). Thus, for our case, the differential privacy definition can be reformulated as:

$$\forall_{D' \in \mathcal{D}} \forall_{x \in D} \quad \mathrm{e}^{-\epsilon} \leq \frac{\Pr[x \in D']}{\Pr[x \in D]} \leq \mathrm{e}^{\epsilon} \,, \tag{3.1}$$

where $\Pr[x \in D]$ and $\Pr[x \in D']$ are estimated using the system's responses to the user's images. Lower values of ϵ indicate better privacy. Our goal is to establish the ϵ_0 value for the system with an unmodified database and then compare it with the ϵ_L value for the system with images modified using LowKey. Generally, we expect the ϵ to be mostly influenced by the user x' missing in $D' \in \mathcal{D}$, since $\Pr[x' \in D]$ should be relatively high (close to 1) and $\Pr[x' \in D']$ low (close to 0). We perform the evaluation using our dataset with 100 classes, with 10 images per person in the embedding database and 30 images per person in the evaluation set. To estimate $\Pr[x \in d]$ for user x and database d, we use the proportion of correctly identified images of that person by the system with that database. We examine all databases from $\mathcal{D}$, but for clarity in the descriptions below, by D' we mean the database that leads to the highest ϵ value. Additionally, x' denotes the user that is absent in D' but present in D.

FaceNet

For the FaceNet model and the unmodified dataset, the estimated probabilities for users present in both D and D' range from a minimum of 0.533 to a maximum of 1.0, with a mean probability of 0.936. The probability $\Pr[x' \in D]$ for x' is 1.0, and $\Pr[x' \in D']$ is 0.033. For the LowKey-modified dataset, the estimated probabilities for users in X range from 0.367 to 1.0, with a mean probability of 0.696. The probability $\Pr[x' \in L(D)]$ for x' is 0.933, and $\Pr[x' \in L(D')]$ remains 0.033. From these results, we can conclude that:

$$\epsilon_0 = \max_{D' \in \mathcal{D}} \left[\max_{x \in D} \left| \log \left(\frac{\Pr[x \in D']}{\Pr[x \in D]} \right) \right| \right] = 3.4012\ , \tag{3.2}$$

$$\epsilon_L = \max_{D' \in \mathcal{D}} \left[\max_{x \in D} \left| \log \left(\frac{\Pr[x \in L(D')]}{\Pr[x \in L(D)]} \right) \right| \right] = 3.3322\ . \tag{3.3}$$

ArcFace

For the ArcFace model, the estimated probabilities for users in both D and D' range from 0.7 to 1.0, with a mean probability of 0.952. The probability $\Pr[x' \in D]$ for x' is 1.0, and $\Pr[x' \in D']$ is 0.033. When using the LowKey-modified dataset, the probabilities for users in X range from 0.3 to 0.967, with a mean probability of 0.773. The probability $\Pr[x' \in L(D)]$ for x' is 0.867, and $\Pr[x' \in L(D')]$ remains 0.033. Based on these observations, we conclude that:

$$\epsilon_0 = \max_{D' \in \mathcal{D}} \left[\max_{x \in D} \left| \log \left(\frac{\Pr[x \in D']}{\Pr[x \in D]} \right) \right| \right] = 3.4012\ , \tag{3.4}$$

$$\epsilon_L = \max_{D' \in \mathcal{D}} \left[\max_{x \in D} \left| \log \left(\frac{\Pr[x \in L(D')]}{\Pr[x \in L(D)]} \right) \right| \right] = 3.2581\ . \tag{3.5}$$

GhostFaceNet

For the GhostFaceNet model, the estimated probabilities for users in both D and D' span from 0.5 to 1.0, with a mean probability of 0.931. The probability $\Pr[x' \in D]$ for x' is 1.0, and $\Pr[x' \in D']$ is 0.033. With the LowKey-modified dataset, the probabilities for users in X range from 0.267 to 0.967, with a mean probability of 0.754. The probability $\Pr[x' \in L(D)]$ for x' is 0.800, and $\Pr[x' \in L(D')]$ remains 0.033. From these findings, we deduce that:

$$\epsilon_0 = \max_{D' \in \mathcal{D}} \left[\max_{x \in D} \left| \log \left(\frac{\Pr[x \in D']}{\Pr[x \in D]} \right) \right| \right] = 3.4012 \;, \tag{3.6}$$

$$\epsilon_L = \max_{D' \in \mathcal{D}} \left[\max_{x \in D} \left| \log \left(\frac{\Pr[x \in L(D')]}{\Pr[x \in L(D)]} \right) \right| \right] = 3.1781 \;. \tag{3.7}$$

These results indicate that the LowKey modification consistently reduces the ϵ value across different models: from 3.4012 to 3.3322 for FaceNet, from 3.4012 to 3.2581 for ArcFace, and from 3.4012 to 3.1781 for GhostFaceNet. This reduction suggests that the LowKey modification enhances the differential privacy of the facial recognition system, making it more resilient to privacy breaches. The lower ϵ_L values reflect stronger privacy guarantees, indicating that the modified system is better at obscuring the presence or absence of an individual in the dataset. However, we must bear in mind that the epsilon has been lowered due to the decrease in the accuracy of the system. It may be beneficial to also increase the $\Pr[x' \in L(D')]$, which would ideally balance maintaining accuracy while still enhancing privacy. This means that while LowKey has successfully reduced ϵ, further improvements could focus on minimizing the trade-off between privacy and accuracy. By enhancing the system's ability to obscure individual presence without significantly compromising correct identification rates, we can achieve an even more robust and privacy-preserving facial recognition system.

Sensitivity Analysis

Introduced in 1.1.3 concept of ℓ_1-sensitivity helps us understand the effect each modification has on the embedding space. Specifically, we aim to determine how the absence of a particular class (a person in the dataset) affects the overall accuracy of the system. Increased sensitivity, which in this context refers to the variability of accuracy, can indicate the amount of distortion in the embedding space introduced by the modifications. High sensitivity suggests that the embedding space is significantly altered when a class is removed, potentially leading to decreased system robustness. Although it will be necessary to explore larger datasets and focus separately on accuracy for each class to draw more definitive conclusions, the data in Table 3.1 suggests that some of the modifications, including LowKey, indeed increase sensitivity. This indicates that these methods introduce some form of distortion to the embedding space, affecting the system's stability. Further studies with larger and more diverse datasets are essential to validate these results comprehensively.

Table 3.1. Sensitivity analysis of accuracy for 100 classes.

Modification	**FaceNet**	**ArcFace**	**GhostFaceNet**
none	0.0032	0.0023	0.0031
blur	0.0044	0.0037	0.0075
permute	0.0025	0.0024	0.0033
style transfer	0.0045	0.0022	0.0124
fawkes (low)	0.0047	0.0041	0.0041
fawkes (mid)	0.0046	0.0047	0.0040
fawkes (high)	0.0043	0.0138	0.0047
lowkey	0.0052	0.0036	0.0126

Analyzing sensitivity across different classes individually can provide deeper insights into how each modification affects specific parts of the dataset. This nuanced understanding will be crucial for developing facial recognition systems that are both privacy-preserving and reliable in real-world applications.

Summary

The primary objective of this work was to develop a facial recognition system that operates on modified embeddings to enhance security while maintaining usability. We focused on image modifications as a means to achieve this goal, exploring how such modifications can serve as an additional layer of privacy. To achieve this, we provided a comprehensive theoretical background and a literature review, underscoring the pivotal role of embeddings in modern deep learning and presenting differential privacy as a framework for measuring users' privacy. We analyzed current trends in facial recognition, highlighting modern architectures and loss functions designed to create efficient models for embedding extraction. Several models, including FaceNet [26], ArcFace [8], and GhostFaceNet [2], were used in our experiments to ensure reproducibility across different systems. We utilized the FaceScrub dataset to create the databases and evaluation sets. Various image modifications were tested, ranging from basic techniques to more advanced methods, such as data poisoning techniques like Fawkes [27] and LowKey [6]. The LowKey modification showed particularly promising results, achieving significantly better accuracy on modified images compared to original ones for all tested models. For instance, the ArcFace model achieved an accuracy of 74% on the evaluation set with modified images and 25% for unmodified images, suggesting that the applied modification can offer an additional layer of privacy by making the embeddings significantly less useful to potential attackers in the event of a data leakage. Our differential privacy analysis underscored the potential of the LowKey modification to obscure the presence or absence of an individual in the dataset, thereby enhancing privacy. The LowKey modification demonstrated a reduction in the ϵ value across every tested architecture, reflecting stronger differential privacy guarantees. Specifically, for a 100-class database, the LowKey modification provided 3.1781-differential privacy for GhostFaceNet, 3.2581-differential privacy for ArcFace, and 3.3322-differential privacy for FaceNet. However, this enhancement in privacy came with a trade-off in system accuracy, as the modification affected the stability and robustness of the facial recognition system. Additionally, we performed a sensitivity analysis to examine variations in the system's accuracy in the absence of a single person from the database. The increased sensitivity in modified databases suggested that the modifications introduced some disturbance in the embedding space. Overall, our experiments demonstrate the feasibility of balancing privacy and accuracy in facial recognition systems. While further refinement is necessary to ensure these systems remain reliable and effective, our results represent a significant step toward developing privacy-preserving facial recognition technologies.

Final Thoughts and Future Perspectives

We achieved the thesis's goal using the LowKey modification. Although it does not meet the ideal scenario, where the system works perfectly on modified embeddings and fails completely on originals, it can serve as a cornerstone for developing better approaches. The effectiveness of LowKey adversarial perturbations introduces new research possibilities, such as loosening the constraint on perceptual difference or replacing it with another metric. Another important research direction is examining the modifications' influence on the embedding space. To perform this properly, it may be necessary to investigate larger datasets and focus separately on the accuracy for each individual. By enhancing the system's ability to obscure individual presence without significantly compromising correct identification rates, we can achieve an even more robust and privacy-preserving facial recognition system. In conclusion, we believe that our work contributes to the field of facial recognition by offering an additional layer of security. Future research should aim to refine these modification techniques to minimize the trade-off between privacy and accuracy, ensuring that facial recognition systems remain both effective and secure in real-world applications.

Bibliography

[1] Aggarwal, D., Zhou, J., Jain, A.K., *Fedface: Collaborative learning of face recognition model*, w: *International IEEE Joint Conference on Biometrics, IJCB 2021, Shenzhen, China, August 4-7, 2021* (IEEE, 2021), p. 1–8.

[2] Alansari, M., Hay, O.A., Javed, S., Shoufan, A., Zweiri, Y., Werghi, N., *Ghostfacenets: Lightweight face recognition model from cheap operations*, IEEE Access. 2023, vol. 11, p. 35429–35446.

[3] Best-Rowden, L., Jain, A.K., *Longitudinal study of automatic face recognition*, IEEE Trans. Pattern Anal. Mach. Intell. 2018, vol. 40, 1, p. 148–162.

[4] Bhuiyan, J., *Clearview ai uses your online photos to instantly id you. that's a problem, lawsuit says*, `https://www.latimes.com/business/technology/story/2021-03-09/clearview-ai-lawsuit-privacy-violations`. 2021. [Online; accessed 12.06.2024].

[5] Chen, S., Liu, Y., Gao, X., Han, Z., *Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices*, w: *Biometric Recognition*, ed. by J. Zhou, Y. Wang, Z. Sun, Z. Jia, J. Feng, S. Shan, K. Ubul, Z. Guo (Springer International Publishing, Cham, 2018), p. 428–438.

[6] Cherepanova, V., Goldblum, M., Foley, H., Duan, S., Dickerson, J.P., Taylor, G., Goldstein, T., *Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition*, w: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021* (OpenReview.net, 2021).

[7] Chollet, F., *Xception: Deep learning with depthwise separable convolutions*, w: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, Los Alamitos, CA, USA, 2017), p. 1800–1807.

[8] Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I., Zafeiriou, S., *Arcface: Additive angular margin loss for deep face recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022, vol. 44, 10, p. 5962–5979.

[9] Dhar, P., Bansal, A., Castillo, C.D., Gleason, J., Phillips, P.J., Chellappa, R., *How are attributes expressed in face dcnns?*, w: *15th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2020, Buenos Aires, Argentina, November 16-20, 2020* (IEEE, 2020), p. 85–92.

[10] Dwork, C., Roth, A., *The algorithmic foundations of differential privacy*, Found. Trends Theor. Comput. Sci. 2014, vol. 9, 3–4, p. 211–407.

[11] Federal Bureau of Investigation, *Fbi announces contract award for next generation identification system*, `https://archives.fbi.gov/archives/news/pressrel/press-releases/`

`fbi-announces-contract-award-for-next-generation-identification-system`. 2008. [Online; accessed 12.06.2024].

[12] Gatys, L.A., Ecker, A.S., Bethge, M., *Image style transfer using convolutional neural networks*, w: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), p. 2414–2423.

[13] Gomez-Barrero, M., Maiorana, E., Galbally, J., Campisi, P., Fiérrez, J., *Multi-biometric template protection based on homomorphic encryption*, Pattern Recognit. 2017, vol. 67, p. 149–163.

[14] Goodfellow, I., Bengio, Y., Courville, A., *Deep Learning* (MIT Press, 2016). `http://www.deeplearningbook.org`.

[15] Goodfellow, I.J., Shlens, J., Szegedy, C., *Explaining and harnessing adversarial examples*, w: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Y. Bengio, Y. LeCun (2015).

[16] Han, K., Wang, Y., Tian, Q., Guo, J., Xu, C., Xu, C., *Ghostnet: More features from cheap operations*, w: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, Los Alamitos, CA, USA, 2020), p. 1577–1586.

[17] He, K., Zhang, X., Ren, S., Sun, J., *Deep residual learning for image recognition*, w: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016* (IEEE Computer Society, 2016), p. 770–778.

[18] Hermans, A., Beyer, L., Leibe, B., *In defense of the triplet loss for person re-identification*, CoRR. 2017, vol. abs/1703.07737.

[19] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., *Mobilenets: Efficient convolutional neural networks for mobile vision applications*, CoRR. 2017, vol. abs/1704.04861.

[20] Jain, A.K., Deb, D., Engelsma, J.J., *Biometrics: Trust, but verify*, IEEE Trans. Biom. Behav. Identity Sci. 2022, vol. 4, 3, p. 303–323.

[21] Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L., *Sphereface: Deep hypersphere embedding for face recognition*, w: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE Computer Society, Los Alamitos, CA, USA, 2017), p. 6738–6746.

[22] Mai, G., Cao, K., Yuen, P.C., Jain, A.K., *On the reconstruction of face images from deep face templates*, IEEE Trans. Pattern Anal. Mach. Intell. 2019, vol. 41, 5, p. 1188–1202.

[23] Mohri, M., Rostamizadeh, A., Talwalkar, A., *Foundations of Machine Learning*, 2 ed., Adaptive Computation and Machine Learning (MIT Press, Cambridge, MA, 2018).

[24] Radiya-Dixit, E., Hong, S., Carlini, N., Tramèr, F., *Data poisoning won't save you from facial recognition*, w: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022* (OpenReview.net, 2022).

[25] Raghavendra, R., Busch, C., *Presentation attack detection methods for face recognition systems: A comprehensive survey*, ACM Comput. Surv. 2017, vol. 50, 1, p. 8:1–8:37.

[26] Schroff, F., Kalenichenko, D., Philbin, J., *Facenet: A unified embedding for face recognition and clustering*, w: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015* (IEEE Computer Society, 2015), p. 815–823.

[27] Shan, S., Wenger, E., Zhang, J., Li, H., Zheng, H., Zhao, B.Y., *Fawkes: Protecting privacy against unauthorized deep learning models*, w: *29th USENIX Security Symposium, USENIX Security 2020, August 12-14, 2020*, ed. by S. Capkun, F. Roesner (USENIX Association, 2020), p. 1589–1604.

[28] Sharma, R., *The Making of Aadhaar: World's Largest Identity Platform* (Rupa, 2020).

[29] Simonyan, K., Zisserman, A., *Very deep convolutional networks for large-scale image recognition*, w: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, ed. by Y. Bengio, Y. LeCun (2015).

[30] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., *Going deeper with convolutions*, w: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015* (IEEE Computer Society, 2015), p. 1–9.

[31] Taigman, Y., Yang, M., Ranzato, M., Wolf, L., *Deepface: Closing the gap to human-level performance in face verification*, w: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), p. 1701–1708.

[32] U.S. Department of Homeland Security, *Centralized area video surveillance system*, `https://www.dhs.gov/publication/centralized-area-video-surveillance-system`. 2013. [Online; accessed 12.06.2024].

[33] Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W., *Cosface: Large margin cosine loss for deep face recognition*, w: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (Computer Vision Foundation / IEEE Computer Society, 2018), p. 5265–5274.

[34] Wen, Y., Zhang, K., Li, Z., Qiao, Y., *A discriminative feature learning approach for deep face recognition*, w: *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, ed. by B. Leibe, J. Matas, N. Sebe, M. Welling, *Lecture Notes in Computer Science*, vol. 9911 (Springer, 2016), p. 499–515.

[35] Zeiler, M.D., Fergus, R., *Visualizing and understanding convolutional networks*, w: *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, ed. by D.J. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars, *Lecture Notes in Computer Science*, vol. 8689 (Springer, 2014), p. 818–833.

List of Figures

List of Tables

Appendix

A. Detailed Results of Experiments

Table A.1. Detailed results of the Detailed results for the system with 50 classes and FaceNet as an extractor of the embeddings.

Database Modification	Test Modification	EER	Accuracy
none	none	0.046	0.952
none	blur	0.212	0.747
none	block permutation	0.526	0.237
none	Fawkes (low)	0.115	0.874
none	Fawkes (mid)	0.270	0.663
none	Fawkes (high)	0.322	0.565
none	LowKey	0.444	0.383
none	style transfer	0.347	0.517
blur	none	0.132	0.858
blur	blur	0.220	0.726
block permutation	none	0.494	0.285
block permutation	block permutation	0.508	0.282
Fawkes (low)	none	0.106	0.887
Fawkes (low)	Fawkes (low)	0.152	0.830
Fawkes (low)	Fawkes (mid)	0.251	0.697
Fawkes (low)	Fawkes (high)	0.302	0.614
Fawkes (mid)	none	0.232	0.733
Fawkes (mid)	Fawkes (low)	0.228	0.733
Fawkes (mid)	Fawkes (mid)	0.244	0.702
Fawkes (mid)	Fawkes (high)	0.276	0.664
Fawkes (high)	none	0.331	0.602
Fawkes (high)	Fawkes (low)	0.302	0.634
Fawkes (high)	Fawkes (mid)	0.281	0.658
Fawkes (high)	Fawkes (high)	0.279	0.669
LowKey	none	0.421	0.396
LowKey	LowKey	0.277	0.678
style transfer	none	0.349	0.620
style transfer	style transfer	0.355	0.574

Table A.2. Detailed results for the system with 50 classes and ArcFace as an extractor of the embeddings.

Database Modification	**Test Modification**	**EER**	**Accuracy**
none	none	0.034	0.965
none	blur	0.070	0.919
none	block permutation	0.496	0.234
none	Fawkes (low)	0.097	0.896
none	Fawkes (mid)	0.293	0.644
none	Fawkes (high)	0.364	0.547
none	LowKey	0.509	0.264
none	style transfer	0.407	0.381
blur	none	0.051	0.945
blur	blur	0.074	0.911
block permutation	none	0.481	0.283
block permutation	block permutation	0.488	0.353
Fawkes (low)	none	0.063	0.932
Fawkes (low)	Fawkes (low)	0.101	0.890
Fawkes (low)	Fawkes (mid)	0.240	0.726
Fawkes (low)	Fawkes (high)	0.288	0.654
Fawkes (mid)	none	0.198	0.776
Fawkes (mid)	Fawkes (low)	0.198	0.782
Fawkes (mid)	Fawkes (mid)	0.215	0.756
Fawkes (mid)	Fawkes (high)	0.219	0.746
Fawkes (high)	none	0.269	0.663
Fawkes (high)	Fawkes (low)	0.249	0.711
Fawkes (high)	Fawkes (mid)	0.239	0.734
Fawkes (high)	Fawkes (high)	0.227	0.746
LowKey	none	0.508	0.271
LowKey	LowKey	0.206	0.768
style transfer	none	0.283	0.687
style transfer	style transfer	0.445	0.405

Table A.3. Detailed results for the system with 50 classes and GhostFaceNet as an extractor of the embeddings.

Database Modification	Test Modification	EER	Accuracy
none	none	0.069	0.926
none	blur	0.108	0.881
none	block permutation	0.509	0.259
none	Fawkes (low)	0.109	0.881
none	Fawkes (mid)	0.207	0.760
none	Fawkes (high)	0.257	0.701
none	LowKey	0.475	0.302
none	style transfer	0.241	0.703
blur	none	0.089	0.904
blur	blur	0.134	0.859
block permutation	none	0.550	0.226
block permutation	block permutation	0.475	0.317
Fawkes (low)	none	0.097	0.895
Fawkes (low)	Fawkes (low)	0.117	0.871
Fawkes (low)	Fawkes (mid)	0.177	0.799
Fawkes (low)	Fawkes (high)	0.205	0.764
Fawkes (mid)	none	0.167	0.811
Fawkes (mid)	Fawkes (low)	0.173	0.808
Fawkes (mid)	Fawkes (mid)	0.181	0.795
Fawkes (mid)	Fawkes (high)	0.198	0.782
Fawkes (high)	none	0.203	0.768
Fawkes (high)	Fawkes (low)	0.193	0.784
Fawkes (high)	Fawkes (mid)	0.180	0.800
Fawkes (high)	Fawkes (high)	0.183	0.795
LowKey	none	0.448	0.326
LowKey	LowKey	0.228	0.744
style transfer	none	0.166	0.815
style transfer	style transfer	0.283	0.673

Table A.4. Detailed results of the Detailed results for the system with 100 classes and FaceNet as an extractor of the embeddings.

Database Modification	**Test Modification**	**EER**	**Accuracy**
none	none	0.061	0.937
none	blur	0.275	0.674
none	block permutation	0.497	0.252
none	Fawkes (low)	0.150	0.834
none	Fawkes (mid)	0.324	0.590
none	Fawkes (high)	0.391	0.485
none	LowKey	0.456	0.348
none	style transfer	0.418	0.429
blur	none	0.166	0.813
blur	blur	0.260	0.672
block permutation	none	0.475	0.288
block permutation	block permutation	0.484	0.261
Fawkes (low)	none	0.116	0.875
Fawkes (low)	Fawkes (low)	0.175	0.804
Fawkes (low)	Fawkes (mid)	0.300	0.629
Fawkes (low)	Fawkes (high)	0.350	0.553
Fawkes (mid)	none	0.254	0.694
Fawkes (mid)	Fawkes (low)	0.278	0.669
Fawkes (mid)	Fawkes (mid)	0.310	0.622
Fawkes (mid)	Fawkes (high)	0.329	0.594
Fawkes (high)	none	0.355	0.560
Fawkes (high)	Fawkes (low)	0.337	0.578
Fawkes (high)	Fawkes (mid)	0.341	0.583
Fawkes (high)	Fawkes (high)	0.329	0.592
LowKey	none	0.447	0.364
LowKey	LowKey	0.309	0.628
style transfer	none	0.367	0.579
style transfer	style transfer	0.381	0.520

Table A.5. Detailed results for the system with 100 classes and ArcFace as an extractor of the embeddings.

Database Modification	**Test Modification**	**EER**	**Accuracy**
none	none	0.046	0.953
none	blur	0.090	0.900
none	block permutation	0.506	0.224
none	Fawkes (low)	0.129	0.861
none	Fawkes (mid)	0.316	0.604
none	Fawkes (high)	0.371	0.500
none	LowKey	0.506	0.261
none	style transfer	0.450	0.350
blur	none	0.069	0.928
blur	blur	0.089	0.894
block permutation	none	0.520	0.246
block permutation	block permutation	0.487	0.334
Fawkes (low)	none	0.086	0.910
Fawkes (low)	Fawkes (low)	0.138	0.850
Fawkes (low)	Fawkes (mid)	0.266	0.686
Fawkes (low)	Fawkes (high)	0.322	0.592
Fawkes (mid)	none	0.220	0.743
Fawkes (mid)	Fawkes (low)	0.227	0.738
Fawkes (mid)	Fawkes (mid)	0.254	0.704
Fawkes (mid)	Fawkes (high)	0.267	0.689
Fawkes (high)	none	0.336	0.578
Fawkes (high)	Fawkes (low)	0.310	0.627
Fawkes (high)	Fawkes (mid)	0.295	0.664
Fawkes (high)	Fawkes (high)	0.270	0.689
LowKey	none	0.534	0.250
LowKey	LowKey	0.227	0.740
style transfer	none	0.330	0.615
style transfer	style transfer	0.456	0.380

Table A.6. Detailed results for the system with 100 classes and GhostFaceNet as an extractor of the embeddings.

Database Modification	**Test Modification**	**EER**	**Accuracy**
none	none	0.069	0.923
none	blur	0.123	0.861
none	block permutation	0.497	0.251
none	Fawkes (low)	0.115	0.868
none	Fawkes (mid)	0.227	0.726
none	Fawkes (high)	0.269	0.663
none	LowKey	0.495	0.284
none	style transfer	0.261	0.660
blur	none	0.103	0.887
blur	blur	0.154	0.832
block permutation	none	0.468	0.275
block permutation	block permutation	0.488	0.290
Fawkes (low)	none	0.098	0.888
Fawkes (low)	Fawkes (low)	0.135	0.848
Fawkes (low)	Fawkes (mid)	0.205	0.761
Fawkes (low)	Fawkes (high)	0.240	0.713
Fawkes (mid)	none	0.182	0.784
Fawkes (mid)	Fawkes (low)	0.182	0.785
Fawkes (mid)	Fawkes (mid)	0.203	0.765
Fawkes (mid)	Fawkes (high)	0.222	0.745
Fawkes (high)	none	0.231	0.716
Fawkes (high)	Fawkes (low)	0.225	0.730
Fawkes (high)	Fawkes (mid)	0.219	0.746
Fawkes (high)	Fawkes (high)	0.226	0.741
LowKey	none	0.534	0.250
LowKey	LowKey	0.255	0.704
style transfer	none	0.209	0.763
style transfer	style transfer	0.288	0.643

B. Summary in Polish

Głównym celem pracy było opracowanie systemu rozpoznawania twarzy, który działa na zmodyfikowanych zanurzeniach (ang. embeddings) w celu zwiększenia bezpieczeństwa użytkowników przy jednoczesnym zachowaniu skuteczności. Konkretnie skupiliśmy się na zanurzeniach odpowiednio zmodyfikowanych zdjęć. Przedstawiliśmy niezbędne zaplecze teoretyczne i przegląd literatury, podkreślając kluczową rolę zanurzeń we współczesnym uczeniu głębokim oraz przedstawiając prywatność różnicową (ang. differential privacy) jako sposób na zapewnienie prywatności użytkowników. Przeanalizowaliśmy obecne trendy w dziedzinie rozpoznawania twarzy, przedstawiając nowoczesne architektury i funkcje strat zaprojektowane do tworzenia efektywnych modeli. W naszych eksperymentach wykorzystaliśmy modele takie jak FaceNet [26], ArcFace [8] i GhostFaceNet [2], oraz bazę danych FaceScrub. Przetestowaliśmy różne modyfikacje obrazów, od podstawowych metod po bardziej zaawansowane techniki zatruwania danych (ang. data poisoning), takie jak Fawkes [27] i LowKey [6]. Modyfikacja LowKey dostarczyła szczególnie obiecujących wyników, pozwalając osiągnąć znacznie lepszą dokładność na zmodyfikowanych obrazach w porównaniu do oryginałów dla wszystkich testowanych architektur. System wykorzystujący model ArcFace oraz zmodyfikowaną przy użyciu LowKey bazę zdjęć, osiągnął dokładność 74% na danych zmodyfikowanych i 25% na oryginalnych. Oznacza to, iż zastosowana modyfikacja może stanowić dodatkową warstwę prywatności, czyniąc zanurzenia znacznie mniej użytecznymi dla potencjalnych atakujących w przypadku wycieku danych. Nasza analiza prywatności różnicowej (ang. differential privacy) podkreśliła potencjał modyfikacji LowKey do zwiększania prywatności użytkowników. Dla bazy obejmującej 100 klas, modyfikacja LowKey zapewniła prywatność różnicową na poziomie 3.1781 dla GhostFaceNet, 3.2581 dla ArcFace i 3.3322 dla FaceNet. Zwiększenie prywatności wiązało się jednak z kompromisem w zakresie dokładności systemu. Dodatkowo przeprowadziliśmy analizę wrażliwości, aby zbadać stopień zaburzeń w przestrzeni zanurzeń wywołanych przez modyfikacje. Reasumując, nasze eksperymenty demonstrują możliwość zbalansowania prywatności i dokładności w systemach rozpoznawania twarzy. Chociaż konieczne jest dalsze udoskonalanie tych metod, aby zapewnić ich niezawodność i skuteczność, nasze wyniki stanowią istotny krok naprzód w kierunku rozwoju systemów rozpoznawania twarzy zapewniających prywatność użytkowników.

Ostateczne wnioski i perspektywy

Cel pracy został osiągnięty wykorzystując modyfikację LowKey. Chociaż nie spełnia ona idealnego scenariusza, w którym system działa perfekcyjnie na zmodyfikowanych zanurzeniach i całkowicie zawodzi w przypadku oryginałów, może stanowić fundament do opracowywania lepszych podejść. Skuteczność zakłóceń metody LowKey otwiera nowe możliwości badawcze, przykładowo w kierunku zastąpienia ograniczenia percepcyjnego inną metryką. Kolejnym ważnym kierunkiem badań jest zbadanie wpływu modyfikacji na przestrzeń zanurzeń. Aby to zrobić właściwie, może być konieczne przeanalizowanie większych zbiorów danych i koncentracja na zmianie dokładności każdej klasy. Podsumowując, uważamy, że nasza praca wnosi istotny wkład w dziedzinę rozpoznawania twarzy, oferując dodatkową warstwę bezpieczeństwa. Przyszłe badania powinny dążyć do udoskonalania tych technik modyfikacji, w celu zapewnienia wysokiego poziomu skuteczności oraz prywatności w nowoczesnych systemach rozpoznawania twarzy.