AnonFACES: Anonymizing Faces Adjusted to Constraints on Efficacy and Security

Citation for the original published paper (version of record):

Le, M., Khan, M., Tsaloli, G. et al (2020)
AnonFACES: Anonymizing Faces Adjusted to Constraints on Efficacy and Security
WPES 2020, Proceedings of the 19th Workshop on Privacy in the Electronic Society: 87-100
http://dx.doi.org/10.1145/3411497.3420220

(article starts on next page)

AnonFACES: Anonymizing Faces Adjusted to Constraints on Efficacy and Security

Minh-Ha Le
Linköping University
Linköping, Sweden
le.minh.ha@liu.se

Md Sakib Nizam Khan
KTH Royal Institute of Technology
Stockholm, Sweden
msnkhan@kth.se

Georgia Tsaloli
Chalmers University of Technology
Gothenburg, Sweden
tsaloli@chalmers.se

Niklas Carlsson
Linköping University
Linköping, Sweden
niklas.carlsson@liu.se

Sonja Buchegger
KTH Royal Institute of Technology
Stockholm, Sweden
buc@kth.se

ABSTRACT

Image data analysis techniques such as facial recognition can threaten individuals' privacy. Whereas privacy risks often can be reduced by adding noise to the data, this approach reduces the utility of the images. For this reason, image de-identification techniques typically replace directly identifying features (*e.g.,* faces, car number plates) present in the data with synthesized features, while still preserving other non-identifying features. As of today, existing techniques mostly focus on improving the naturalness of the generated synthesized images, without quantifying their impact on privacy. In this paper, we propose the first methodology and system design to quantify, improve, and tune the privacy-utility trade-off, while simultaneously also improving the naturalness of the generated images. The system design is broken down into three components that address separate but complementing challenges. This includes a two-step cluster analysis component to extract low-dimensional feature vectors representing the images (embedding) and to cluster the images into fixed-sized clusters. While the importance of good clustering mostly has been neglected in previous work, we find that our novel approach of using low-dimensional feature vectors can improve the privacy-utility trade-off by better clustering similar images. The use of these embeddings has been found particularly useful when wanting to ensure high naturalness and utility of the synthetically generated images. By combining improved clustering and incorporating StyleGAN, a state-of-the-art Generative Neural Network, into our solution, we produce more realistic synthesized faces than prior works, while also better preserving properties such as age, gender, skin tone, or even emotional expressions. Finally, our iterative tuning method exploits non-linear relations between privacy and utility to identify good privacy-utility trade-offs. We note that an example benefit of these improvements is that our solution allows car manufacturers to train their autonomous vehicles while complying with privacy laws.

WPES'20, November 9, 2020, Virtual Event, USA
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8086-7/20/11.
https://doi.org/10.1145/3411497.3420220

KEYWORDS

privacy, k-anonymity, image de-identification

ACM Reference Format:
Minh-Ha Le, Md Sakib Nizam Khan, Georgia Tsaloli, Niklas Carlsson, and Sonja Buchegger. 2020. AnonFACES: Anonymizing Faces Adjusted to Constraints on Efficacy and Security. In *19th Workshop on Privacy in the Electronic Society (WPES'20), November 9, 2020, Virtual Event,USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3411497.3420220

1 INTRODUCTION

In recent years, there has been an enormous increase in the production of video and image data. Widespread use of handheld devices (*e.g.,* smartphones, tablets, digital cameras), surveillance devices (*e.g.,* security cameras), and many other factors have contributed to this global trend. Moreover, recent technologies such as self-driving vehicles rely heavily on machine learning technologies being applied on the captured image and video data to operate autonomously, enhance driving performance, user experiences, or to in other ways make our lives easier. In these usage scenarios, the captured data is analyzed with respect to demographic attributes of the depicted people.

Image and video data, however, are highly privacy-sensitive as they contain biometric or uniquely identifying information of individuals (data subjects). To analyze or share such data while complying with laws such as the GDPR [5], the data owner (e.g., the car manufacturer) or data controller (e.g., the data analysts) need to remove information from the data to de-identify the data subjects. To achieve this goal, de-identification techniques generally try to remove identity-related information in such a way that not only humans but also machines cannot recognize the identity of any individual present in the data. Furthermore, due to its critical role in analyzing data from the aforementioned emerging technologies (*e.g.,* training self-driving cars to detect and better interact with pedestrians), it is becoming increasingly important to preserve as much as possible of the utility of original data in the de-identified data. Ideally, the resulting datasets should also have similar properties as the original dataset (e.g., distributions of attributes such as age, gender, skin tone, or even emotional expression should be preserved), generated faces should look natural, and the methodology should be generally applicable for different datasets, without restrictions on how faces are depicted, and allow for easy addition/removal of images.

Some of the most popular image de-identification techniques are from the k-same [22] family. The idea of k-same de-identification is based on the k-anonymity [32] scheme for categorical data, that turns identifiability of individuals into one of a set of k individuals. Generally, k-anonymity based image de-identification techniques first cluster k similar images, then generate a synthesized image by combining the k images from the cluster and, finally, replace each image in the cluster with the same synthesized image. Thus, the privacy and the utility provided by the k-anonymity based de-identification techniques are highly dependent on the value k. However, the k-anonymity based de-identification techniques do not specify how to choose k. Another key factor that has a great impact on the privacy and utility provided is how well we cluster similar images. To increase the utility (decrease the information loss), it is important to cluster similar images into one cluster. When using k-anonymity schemes, typically all such clusters have the same number of elements (*i.e.,* k). While such fixed-size clusters adhere to the privacy bound $1/k$, no prior work has provided techniques to cluster similar images into fixed-size clusters in a way that takes information loss into account or made use of machine-learning-based embedding techniques such as those used here to improve the clustering. In this paper, we introduce a framework capable of reducing the information loss and tuning the privacy-utility trade-off for a given dataset. To evaluate the utility of different de-identification techniques and the relative value of applying different functions within the different modules of our system design, we present example evaluations on two public datasets. To evaluate the utility in terms of information loss, we measure the average Euclidean distances between each image in a cluster and the corresponding synthesized image for that cluster. Furthermore, we study the privacy-utility trade-off, by varying the size of the clusters generated (*i.e.,* k) and calculating the total information loss for each cluster size. Finally, and perhaps most importantly, we demonstrate how access to this trade-off (as provided for the first time by our framework) yields insights and helps us determine the best k value, given a set of specific privacy and utility requirements.

Our Contribution. In summary, our main contributions are:

- Methodology, conceptual framework, and metrics for *quantifying* the privacy-utility trade-off. Our solutions are applied and evaluated on two datasets.
- A solution framework that *improves* the overall privacy-utility trade-off by increasing utility unilaterally with own algorithms (clustering), novel utilization of tools developed for a different purpose (embedding), and use of a suitable off-the-shelf tools (synthetic-face generation).
- A methodology for *tuning* the privacy-utility trade-off by exploiting non-linear relations where increases in privacy (k in k-anonymity) result in little loss of utility.

Overall, the framework is shown to provide desirable *security* properties (k-anonymity and facial recognition resistance), while improving several *efficacy* properties (e.g., utility, naturalness, and generality) compared to related works.

Organization. The rest of the paper is organized as follows. We first review the history and state of the art of image de-identification in Section 2. We then present AnonFACES in Section 3, including the desired properties, the conceptual framework, and detailed descriptions of the instantiated components of the framework. We evaluate the performance and security in Sections 4 and 5 respectively, followed by our conclusions in Section 6.

2 RELATED WORK

The early data anonymization research mostly focused on protecting privacy of categorical data which produced multiple well-known data de-identification techniques. One such technique for de-identifying entries in a relational database was proposed by Sweeney and termed as k-anonymity [32]. Building on k-anonymity, there are other data anonymization techniques for categorical data proposed in the literature: among those l-diversity [21] and t-closeness [16] are the two most popular.

The early face de-identification approaches started with ad-hoc techniques such as black box, blurring, and pixelation [27]. Even though ad-hoc techniques can prevent humans from reidentification of a subject in a de-identified image, they fail to preserve the utility present in the data and are not robust enough to fool the recognition systems [24]. To overcome these issues, the research then shifted towards techniques with provable privacy guarantees. Since differential-privacy techniques share the problems of blurring and pixelation, the focus has been on k-anonymity. Newton *et al.* [24] first proposed the original k-same algorithm based on the k-anonymity model. The original k-same algorithm had limitations in terms of the naturalness of the synthesized images and also in terms of information loss during the de-identification process. Many improvements of the original k-same algorithm have been proposed to overcome these limitations including k-Same-Select [9], k-Same-M [10], k-same-furthest [23], and k-Diff-furthest [31], to name a few. These techniques either use Active Appearance Model (AAM) or Principle Component Analysis (PCA) to construct and preserve the different facial attributes such as age or gender.

The de-identified images produced by k-same based approaches discussed previously lack naturalness. Generative Adversarial Networks (GANs) are recent generative models that can produce natural-looking synthesized images of any given object using adversarial training. This idea was first proposed by Goodfellow *et al.* in [8]. The synthesized images produced by GANs are also visually convincing for the human eye. Since GANs are capable of producing natural-looking synthesized images, they are well suited for de-identification. As a result, multiple works proposed GAN-based image de-identification techniques. One such technique is Privacy-Protective-GAN (PP-GAN) proposed by Wu *et al.* [34]. PP-GAN is designed for face de-identification by adapting GAN with novel verificator and regulator modules. It is capable of retaining structure similarity in the de-identified output based on a single input. Similarly, AnonymousNet [17] extracts facial features for structure but adds noise for GAN-generated images.

k-same-Net proposed by Meden *et al.* [22] is another image de-identification scheme that aims to combine the k-same algorithm with a GAN. Similarly, k-Same-Siamese-GAN proposed by Pan *et al.* [26] is also a GAN based de-identification scheme that combines k-anonymity, GAN, and hyperparameter tuning methods to efficiently train the GAN networks and de-identify images with

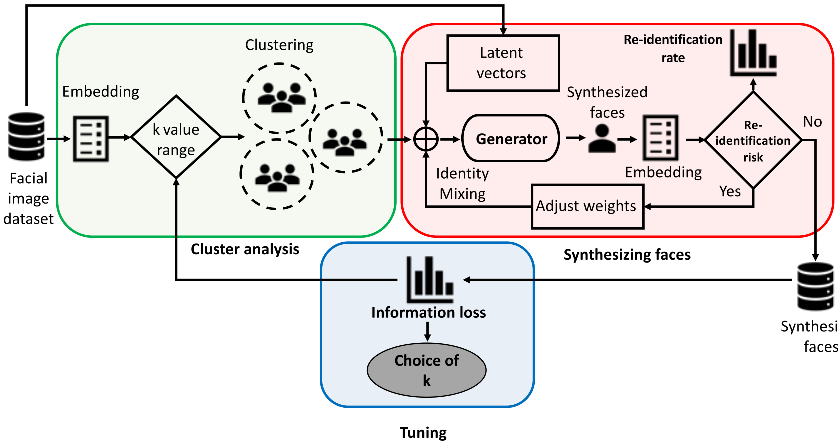

Figure 1: Overview of system model

provable privacy guarantees. Nevertheless, there are still limitations in the GANs used by these techniques. For example, while the k-Same-Siamese-GAN and PP-GAN still lack in naturalness, k-same-Net needs to be re-trained for every new image dataset, which is inefficient for practical use cases.

None of the related works provide the quantification and methodology needed to find good values for k in terms of the privacy-utility trade-off. Due in part to the concrete instantiations for different functions such as clustering and synthetic-face generation, they have lower utility, naturalness, and generality than AnonFACES.

3 ANONFACES COMPONENTS

In this section, we introduce our proposed methodology and system implementation, AnonFACES, to quantify, improve and tune the privacy-utility trade-off in image de-identification. We first identify the desired properties of such a system, and then, after a high-level system overview, explain how AnonFACES is designed to achieve these properties.

3.1 Desirable System Properties

We categorize the desired properties according to whether they relate to security (S) or efficacy (E).

S1: k-anonymity. Increased k means a larger anonymity set and, thus, more privacy.

S2: Facial recognition resistance. Synthesized faces should not match identified faces from the original dataset, and vice versa.

E1: Utility . Non-identifying attributes should be preserved for accuracy of analysis. To generalize to analysis-agnostic cases, the information loss should be low.

E2: Naturalness. Synthesized faces should look like actual faces.

E3: Generality. The methodology should work for different datasets, without restrictions on how faces are depicted (*e.g.*, angles), allow for addition and removal of images, and for different instantiations of functional components.

3.2 System Overview

Our proposed model (Figure 1) consists of the following three main components.

Cluster analysis. The challenge is to generate fixed-size clusters for k-anonymity and at the same time minimize the information loss. To do so, we divide the process into two steps: extraction of low-dimensional feature vectors that represent the images (**embedding**), and **fixed-size clustering** based on these feature vectors.

Synthesizing faces. For synthesizing one face from the k images in a cluster, the first step is **identity mixing**. The resulting vector then is input for the image **generation**. The synthesized image is tested with facial recognition for **risk assessment** of re-identification, with weight adaptation for the identity mixing if need be.

Tuning. Different cluster sizes (k values) are evaluated for their associated **information loss**, to inform the output selection of good **values for** k and the corresponding synthesized faces for each cluster. We define the information loss for a set of de-identified images according to Def. 3.2.

Each of the modules that perform the functions provided by the components can be instantiated in various ways. We describe the different modules and promising instantiations in more detail in the following subsections.

3.3 Cluster Analysis

3.3.1 Definitions. For completeness, we provide, in this section, some definitions necessary to follow our work. First, a person-specific dataset is presented below based on the definition suggested by Newton *et al.* [24].

Definition 3.1 (Person-specific dataset). Let $\mathcal{H}$ be a dataset containing M images, *i.e.*, $\{\mathrm{H}_1, \ldots, \mathrm{H}_M\}$. Then, $\mathcal{H}$ is a person-specific dataset if and only if *(i)* each image $\{\mathrm{H}_i\}_{i\in[M]} \in \mathcal{H}$ relates to only one person and *(ii)* no two images $\mathrm{H}_i, \mathrm{H}_j, \in \mathcal{H}, i \neq j$ relate to the same person.

From now on in this paper, if we do not specify otherwise, the term dataset we use is person-specific dataset. To formalize our utility metric, we shall need the following definition.

Definition 3.2 (Information Loss). Let $\vec{\mathrm{I_1}}, \ldots, \vec{\mathrm{I_N}}$ and $\vec{\mathrm{D_1}}, \ldots, \vec{\mathrm{D_N}}$ be the sets of the vector representation of the original images and corresponding de-identified images with $\vec{\mathrm{I_m}} = (\mathrm{i_{m_1}}, \ldots, \mathrm{i_{m_p}})$ and $\vec{\mathrm{D_m}} = (\mathrm{d_{m_1}}, \ldots, \mathrm{d_{m_p}})$, where $\mathrm{m} \in [N]$ and p is the vectors' dimension. Then, we define information loss to be the average pair-wise Euclidean distance of the de-identified images to original images. More formally, the information loss (IL) is defined as follows:

$$\mathrm{IL} = \frac{\sum_{m=1}^{N} \sqrt{\sum_{l=1}^{p} (\mathrm{i_{m_l}} - \mathrm{d_{m_l}})^2}}{N}$$

3.3.2 Embedding. The goal here is to compress the original image database into condensed and lightweight embeddings which can improve clustering. For embedding, we base our idea on the deep-similarly metric for facial images, which achieves a high level of precision as shown in recent works [11, 30, 33]. In contrast to the related research, which mostly use dimensionality reduction techniques such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA), in AnonFACES, we use Convolutional Neural Network(CNN) based techniques for embedding. Dimensionality reduction techniques are generic techniques for representing high dimensional data into low dimensional space and are not tailored for clustering similar identities, whereas the CNN based embedding techniques are designed for that specific purpose. The embeddings extracted using CNN not only provide insights into the structure of the input dataset but also can help reducing information loss of de-identification.

Two such face embedding techniques are Dlib [14] and FaceNet [30]. They are trained to generate embeddings in such a way that the distance between embeddings of the same identity or similar identities is small and that between different identities is large. Even though we use these face embedding techniques for a different but related purpose, our experiments show that by using the embeddings generated by these techniques it is possible to instead cluster images of *different* identities who have similar features (*e.g.*, age, gender) into one cluster.

3.3.3 Clustering. The goal of the clustering module is to cluster similar images into fixed-size clusters and the existing state-of-the-art clustering techniques are not tailored to do so. Thus, we need to define the clustering criteria and what is clustering accuracy for our work. Our clustering criteria are based on the deep similarity metric where it is expected that given the embeddings, the clustering algorithm should find equivalence classes in which the members in a cluster should have the smallest possible pair-wise distance.

- **Evaluation Metric.** We evaluate the clustering based on a) Pair-wise distance: min, average, and max distance; b) Mean Silhouette scores. While the former provides a view on how pair-wise distances between faces are distributed, the latter measures the coherence and separation of clusters [28].
- **Size Constraint on Clustering.** To ensure that all the clusters adhere to k-anonymity, we need to generate clusters in such a way that each cluster contains at least k elements. In the case of images, it makes sure that at least k facial images are chosen for generating a synthesized face.
- **Hierarchical Partitioning.** Since there is no existing clustering technique to cluster similar images into fixed-size clusters based on the deep similarity metric, we developed our own algorithm. As the embeddings are optimized for comparing pair-wise distance between faces, building a hierarchical tree based on distance matrix is a natural choice. Based on this notion, in our algorithm, we build a hierarchical tree and cut the tree at different thresholds until we find a cluster with at least k members. To ensure that each cluster has at least k members, each cluster is pre-assigned a cluster size. In default mode, all clusters are assigned either a size k or $k + 1$ (when n is not evenly divided by k). The algorithm for this approach is described in Algorithm 1. There are four different linkage criteria one can choose for the hierarchical tree. In particular, considering we are clustering two clusters C_0 and C_1, where the pair-wise distance between the two data points is $\delta_{ij} = |P_i - P_j|$ such as $P_i \in C_0$ and $P_j \in C_1$, then the *single linkage* is defined as $\min(\delta_{ij})$ the *complete linkage* is $\max(\delta_{ij})$ ($\forall P_i \in C_0$ and $\forall P_j \in C_1$). In their simplest form, the *average linkage* is $\sum \delta_{ij}/|C_0||C_1|$, and the *ward linkage* is $\sum \delta_{ij}^2/|C_0||C_1|$ ($\forall P_i \in C_0$ and $\forall P_j \in C_1$), where $|C_0|$ and $|C_1|$ are sizes of the clusters.

Algorithm 1: Hierarchical partitioning

```
Input: $S$: input dataset with $n$ records, $k$: size constraint,
       $link$: linkage criteria
Output: $C$: set of clusters (where size of each cluster $\geq k$)
$m = \lfloor n/k \rfloor$ : number of clusters
$K = \{k_1, k_2, ..., k_m\}$, where $k_i = k + 1$ for $1 \leq i \leq (n \bmod m)$,
  and $k_i = k$ for $(n \bmod m) < i \leq m$
$Z = linkage(S, link)$: Hierarchical tree based on pair-wise
  distance
for $j = 1$ to $m$ do
    $q = \emptyset$
    while $|q| \leq k_j$ do
        $T = cut\_tree(Z, m)$: Cut the tree into $m$ clusters
        $q = \max(T)$: Sub-tree with the biggest size in T
        $q = \{q_1, q_2, ..., q_k\}$: Select the lowest $k$ members in
          the sub-tree
        $m = m - 1$
    $Z = Z - q$: Update the tree
    $m = \lfloor |Z|/k \rfloor$ : Update number of clusters
    $C = C \cup q$
return $C$
```

3.4 Synthesizing faces

The goal of the synthesizing faces component is to generate synthesized images for each cluster by combining all the k images in the corresponding cluster. In our system model, the synthesizing faces component (Figure 1) is comprised of three modules: (1) Identity mixing for combining identity vectors (*i.e.,* latent vectors in the case of GAN networks), (2) Generator for synthesizing image from the mixed latent vector and (3) Re-identification risk assessment for preventing face recognition.

Identity Mixing. The purpose of identity mixing module is to generate one latent vector for each cluster by mixing the latent vectors of all the identities in the corresponding clusters. The mixed latent vector is required as input to the generator. The latent vector for each identity is generated based on the high-level features extracted from the input image. For more details on embedding image to StyleGAN's latent space please refer to [1, 2]. Assuming we are mixing identities of a cluster with k members, each identity is represented by a latent vector LV_i, $1 \le i \le k$. The mixing equation is formed as:

$$LV_{mix} = \Big(\sum_{i=1}^{k} LV_i w_i \Big) \Big/ \Big(\sum_{i=1}^{k} w_i \Big), \tag{1}$$

where w_i with $1 \le i \le k$ is the weight value of a latent vector LV_i. Initially, we set LV_{mix} as the mean of all LV_i, in that case: $w_i = 1/k$. (To avoid a deterministic output, the weights can already at this point be universally randomized within margins, as described in Section 3.4 for images with elevated risk of re-identification. We keep the determinism here for reproducibility of the evaluation.) Note that both the embeddings and the latent vectors can be computed from the high-level features, however, they are used for different purposes. The former is for the task of calculating deep similarity metric between faces for clustering, while the latter is input for the image generator.

Generator. The task of the generator module is generating one synthesized image by combining k images. The choice of generator can have a significant impact on the efficacy of the de-identification process. In our system model the generator can be chosen independently. We experimented with three different generators: Active Appearance Models (used in [10, 23, 29, 31]), Up-convolutional neural network [6] (used by k-same-net [22]), and StyleGAN[12]. In the end, we chose StyleGAN due to its flexibility and performance. One of the key benefits of StyleGAN is that it can generate synthesized images of any unknown identities with a pre-trained network, hence it does not need to be re-trained for every new dataset. In contrast, k-same-net can only generate synthesized images of the identities that it is trained on. Thus, it needs to be re-trained for every new image and every new dataset, which is inefficient. In addition, StyleGAN also provides us the flexibility to control the non-identity related features such as age, gender, emotion, skin color, camera angle, lighting, etc. of the generated synthesized image which is not possible in k-same-net without extra additional manual-parameters (*e.g.,* face expression: happy, fear, sad). Lastly, the naturalness of the synthesized images generated by StyleGAN is better than the counterpart.

Re-identification Risk Assessment. One of the problems while using k-same family algorithms is that there is a possibility that the synthesized face is biased towards one or more identities among k original identities. In an ideal scenario, if we replace k identities with the same synthesized one then the re-identification probability is $1/k$. However, regardless of the choice of generator, there is still bias while synthesizing an image. To tackle this, we introduce a module in our system model for assessing and adjusting the re-identification risk. In our case, we can easily measure the similarity distance from the synthesized identity to the original ones. First, using the similarity distances, we identify whether a synthesized image is below a certain face-recognition threshold distance (*e.g.,* 0.6 in the case of Dlib-based face recognition; this varies for other techniques) from any original identity. Once we detect such a risk, the weights of the identities that are at risk are adjusted and the mixed latent vector is re-calculated by Equation (1).

In particular, assuming that we are generating a synthesized image D_i, $1 \le i \le \lfloor n/k \rfloor$ for a cluster C_i with k members $\{m_1, \ldots, m_k\}$, we first calculate the distances $\{\gamma_1, \ldots, \gamma_k\}$ from D_i to $\{m_1, \ldots, m_k\}$. By comparing the distances to the re-identification threshold, we detect identities that are at risk and we denote their set as $R = \{m_{r_1}, \ldots, m_{r_u}\}$, with u being the size of the set of identities under the threshold. The re-identification rate for this cluster is $Q_i = u/k$. The weights of identities in R are reduced by a factor $\beta \in (0, 1)$, *i.e.,* $\{w_{r_1} - \beta, \ldots, w_{r_u} - \beta\}$. With these new weights, the mixed latent vector is updated and a new synthesized image is generated. This is repeated until $u = 0$ and $R = \{\}$.

3.5 Tuning

The process of image de-identification results in a trade-off between privacy and utility. The relation between privacy and utility in the case of image de-identification is nonlinear. Thus, to exploit this relation and find good points in the trade-off between privacy and utility, we repeat the de-identification process for different values of privacy (the value k) and quantify the corresponding utility (information loss). Based on the quantification, we look for nonlinear effects to get recommendations for good values of k that add little information loss. We exemplify this process and other ways of finding a good k on two different datasets in the evaluation.

4 PERFORMANCE EVALUATION

4.1 Datasets and Experiment Environment

To evaluate AnonFACES, we use the Microsoft Azure Virtual Machine with a Standard NC6 configuration: E5-2690v3 Xeon CPU, Tesla K80 GPU, and 56GB of RAM. Our code is available in a GitHub repository[1]. For the datasets, we use two publicly available datasets of face images, *i.e.,* Radboud Faces Database (RaFD) [15] and Large-scale CelebFaces Attributes (CelebA) Dataset [20].

- **RaFD:** The RaFD dataset contains high-quality images of 67 subjects with eight different facial expressions (*i.e.,* anger, disgust, fear, happiness, sadness, surprise, contempt and neutral) per subject. Furthermore, for each facial expression, each subject is captured under three different gaze directions and from five camera angles.
- **CelebA:** The CelebA dataset is a popular large-scale dataset containing over 200k celebrity images, each with 40 attribute annotations. The diverse images in this dataset cover 10,177 different identities, large pose variations, and include a rich variation in background clutter.

Regarding the generator, we use StyleGAN [12]. Incorporating StyleGAN into our experiments is straightforward since pre-trained models are available for the two high-quality datasets FFHQ and CelebHQ [25]. Whereas CelebHQ contains high-quality images of

[1] https://github.com/minha12/AnonFACES

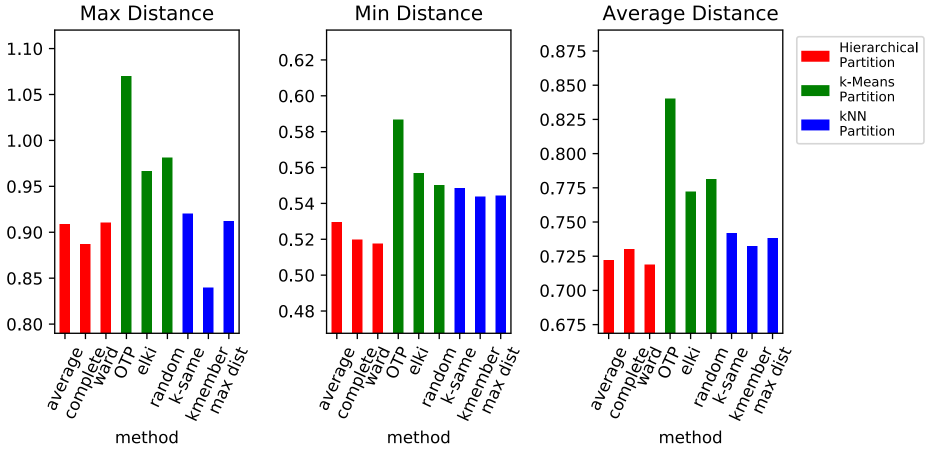

Figure 2: Comparing internal pair-wise distances on CelebA dataset with different algorithms: (Hierarchical Partition): average, complete and ward linkage; (k-Means Partition): OTP [19], elki [3], random method; (kNN Partition): k-same clustering [24], $k$member [4], max dist method. Here, we used $k = 20$.

the same identities as CelebA, the FFHQ contain portrait images of normal people on Flickr.com. For the StyleGAN experiments presented in this section, we used the pre-trained models based on the FFHQ dataset. This allows us to target a more realistic use case in which a network is trained and applied on datasets containing different identities.

4.2 Clustering Evaluation

Besides our proposed hierarchical partitioning algorithm, we implemented other related works and grouped them according to their relative algorithm, *i.e.,* kNN Partition includes k-same clustering [24], $k$member [4], and max dist method - a variance of clustering used in k-same with maximum distance function for selecting initial cluster centroids; k-means Partition includes elki [3] and random method - a variance of elki with random selection as a baseline comparison. To evaluate how well different clustering algorithms group similar images, we calculate the minimum, average, and maximum pair-wise distance within a cluster, as well as the mean Silhouette scores. The results of the experiments performed on CelebA dataset are shown in Figures 2 and 3. The choice of focusing on distance metrics within a cluster rather than cross-cluster metrics is motivated by the observation that the internal distances metrics is what matter most for minimizing the information loss when combining similar faces in a cluster. It matters much less how far apart the faces in different clusters are separated.

Comparing the distance scores of the algorithms (Figure 2), we observe a slight advantage for hierarchical partitioning in many but not all cases, including when the best linkage criteria are used (*i.e.,* when combined with average or ward linkage the hierarchical partitioning performs best on average). Note that for both Figures 2 and 3, we narrow down the y-axis to focus on the changes.

Considering the Silhouette scores (Figure 3), the hierarchical partitioning again stands out as the winner in most cases and is the only partitioning technique that achieves positive scores (again when using average and ward linkage). A positive Silhouette score means that similar faces are well situated in their cluster rather than their nearest-neighbor cluster. These results indicate that for CelebA dataset, hierarchical partitioning is best suited. While the best method may differ for other datasets, the methodology described here can easily be replicated to find the best algorithm also for other datasets. Therefore, our methodology satisfies the desired property for generality (E3).

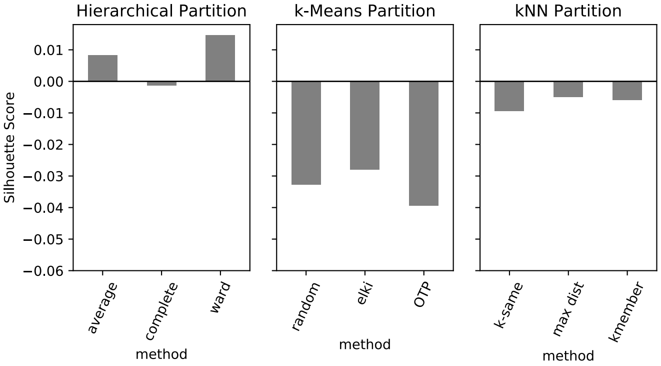

Figure 3: Clustering evaluation: comparing average Silhouette scores on CelebA dataset. Here, we used $k = 20$.

4.3 Information Loss Evaluation

In this section, we experimentally evaluate the improvements provided by AnonFACES as well as how the configurations at each stage of the process affect the privacy-utility trade-off.

To capture the utility of the images generated by AnonFACES, we first show its naturalness preservation capability and then, we use the information loss metric to evaluate the information loss attributed to some of the design choices made in the AnonFACES design. For the latter part, we present a step-by-step analysis, taking into account the information loss associated with three of the main steps: embedding, clustering, and image generation.

Preserving Naturalness (E2). Preserving the naturalness of de-identified image is the main focus of recent works [22, 26, 34] on image de-identification. This aspect is difficult to measure by embedding techniques since it is subjective to human observation.

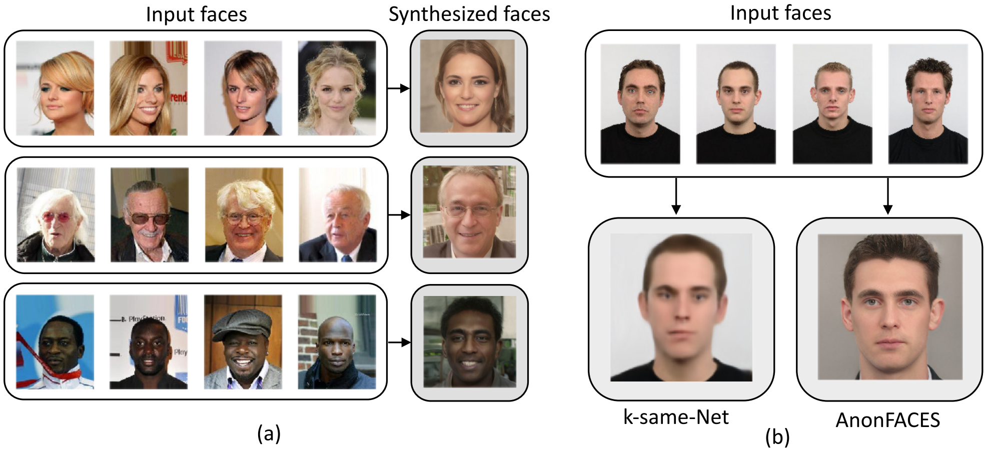

Figure 4: Naturalness preservation: (a) capability of AnonFACES to group similar faces and synthesize faces from different camera angles, age groups, and skin tones (CelebA dataset), (b) comparison of *k*-Same-Net and AnonFACES (RafD dataset)

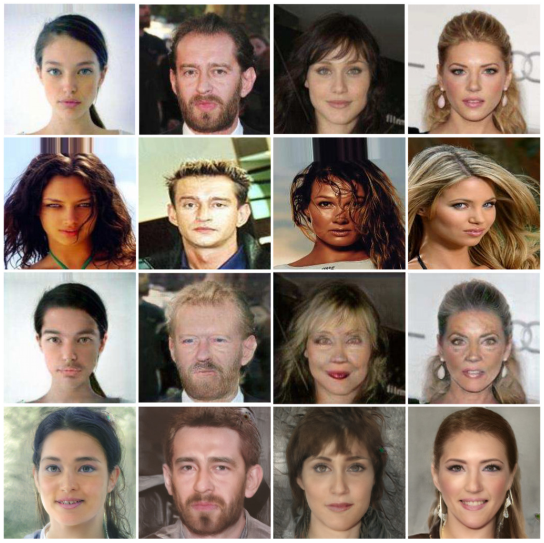

Figure 5: Comparison with AnonymousNet [17]: first row: input images, second row: similar faces found using AnonFACES clustering, third row: results from [17], fourth row: AnonFACES results

Although it has not much meaning for computer vision, some applications have high requirements in this regard, *e.g.,* publishing de-identified image datasets (the client requires that the de-identified images should be indistinguishable from real images). In Figure 4(a), some sample results show that AnonFACES can cluster and synthesize faces from varying input conditions. Figure 4(b) compares representative examples generated by *k*-Same-Net and AnonFACES on the RafD dataset. As shown in the figure, while *k*-Same-Net is unable to preserve much of the details and also could not generate the hair properly, AnonFACES, thanks to instantiating the generator module with StyleGAN, provides better naturalness with a high level of details, fulfilling Property E2.

Figures 5, 6, and 7 provide direct comparisons with three other recent state-of-the-art solutions [17, 26, 29]. Here, we have used images that the other works have used in their papers as input images (top row), identified similar faces using the AnonFACES clustering technique (second row), and then presented the results that from the competing solutions (third row) and those from AnonFACES (fourth row) with the two first images as input, with k=2.

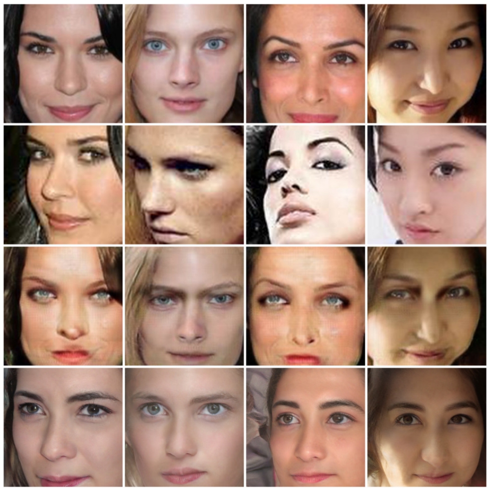

Figure 6: Comparison with k-Same-Siamese-GAN [26]: first row: input images, second row: similar faces found using AnonFACES clustering, third row: results from [26], fourth row: AnonFACES results

Effect of Embeddings Extraction Method on Information Loss (E1). First we show the effect of choice of embedding techniques on the information loss. We compare the embeddings extracted from Dlib, FaceNet, and PCA - a traditional dimensionality-reduction method (used in prior works [22–24]). The experiment is performed using hierarchical partitioning and StyleGAN generator for both of the test cases. The results are shown in Figure 9(a) with k values ranging from 2 to 20. In the figure, we can see a significant difference between Dlib/FaceNet compared to PCA. For example, the information loss when using $k = 3$ with PCA is the same when using $k = 9$ with Dlib. This is a remarkable improvement, since $k = 9$ provides much better privacy than $k = 3$. The improvement

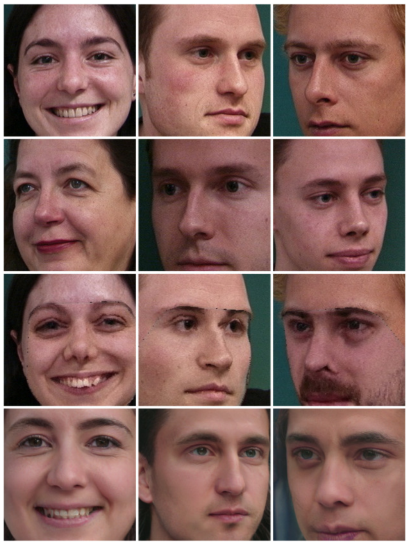

Figure 7: Comparison with Samarzija *et al.* [29], AnonFACES can preserve pose with higher image quality: first row: input images, second row: similar faces found using AnonFACES clustering, third row: results from [29], fourth row: AnonFACES results

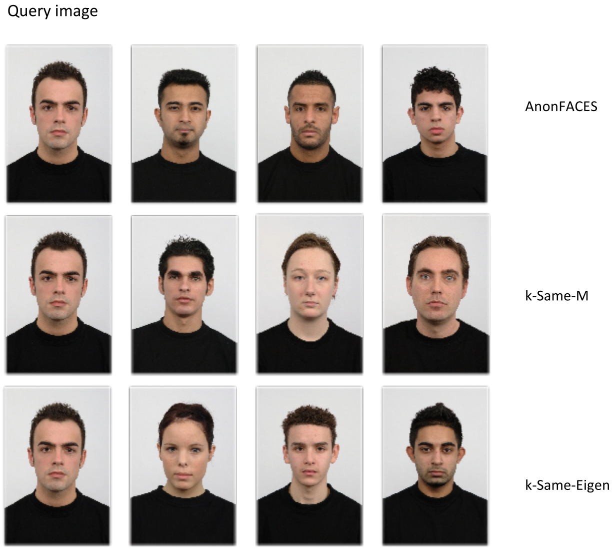

Figure 8: A sample of clusters (k = 4): the first column is the query image; each row contains images from the same cluster as the query. first row: AnonFACES, second row: k-Same-M [10], third row: k-Same [24]

reflects the property (S1) of the desired properties. Regarding recommendation of k value for the Dlib case, the information loss changed significantly with $k < 4$ and $k > 6$ indicating that $k = 5$ or $k = 6$ can be good choices for the given dataset.

Effect of Clustering on Information Loss (E1). This experiment is performed using Dlib embeddings and StyleGAN generator on the CelebA dataset. The result is shown in Figure 9(b). Unlike the results in Section 4.2, the differences between k-means (elki method), kNN (max dist method) and hierarchical partitioning (ward linkage) are noticeable, with the latter slightly outperforming the former for all the tested k values. Furthermore, we observe from the result that the k-means method fluctuates more compare to the other methods. One possible reason behind the fluctuation of k-means can be its random choice of initial centroids. The combination of a good choice embeddings and clustering method also have a visual impact on the clustering result. As shown in Figure 8, we compare the clustering result of AnonFACES based on Dlib embedding and hierarchical partitioning and the related works including k-Same-M [10] and k-Same-Eigen [24]. The k-Same-M is based on Active Appearance Models (AAM), k-NN partitioning (on AAM's features domain) while the k-Same-Eigen is based on PCA embeddings and k-NN partitioning. We can see that k-Same-Eigen mixed different genders and age groups when generating clusters and k-Same-M mixed different genders (although it was better than k-Same-Eigen at grouping people of the same age). In this sample, the result of AnonFACES is better since it can group people of the same age and gender. This is one of the examples showing that even though the improvement of embeddings and clustering may only be slight in terms of numerical results concerning information loss and pair-wise distance (as shown in Figures 2, 3, 9), in terms of attributes grouping the results are noticeable.

Effect of Image Generator on Information Loss (E1). In this experiment, we use Dlib embeddings with hierarchical clustering for comparing StyleGAN, generative up-convolutional neural network [6] (used in k-same-Net [22]), and AAM generator (used in [10, 23, 29, 31]). The up-convolutional network has the main limitation that it requires the identity set used for training and testing to be similar and it only works with a pre-defined number of identities, going against Property E3. Thus, we are only able to perform the experiment with a dataset consisting of a small set of identities. The RafD dataset is chosen this time. As shown in Figure 9(c), except a small different at $k = 2$, StyleGAN has relatively lower information loss than the up-convolutional network and significantly lower than AAM generator in the range of chosen k values. Beside that, the result of the former is less stable compared to the other ones. Note that due to limited space, we narrow down the y-axis of the figures to focus on differences.

Finally, we show a numerical comparison between AnonFACES and the related works in Figure 10. In this figure, we see that AnonFACES found a balance between the information loss and re-identification rate. In most of the cases, we can preserve better information loss while having acceptable re-identification rate below the $1/k$ threshold. We also include the random sampling which is the experiment without any clustering method for the upper-bound estimation. The lower bound depends on the choice of embbeding-extraction method, in case of Dlib embeddings (which is the one we mainly use for our evaluation metrics) it is 0.6. If the result of average information loss is too close to or falls below the threshold, it indicates that the de-identification algorithm is very good at preserving information but also has a higher risk of re-identification, which is the case of k-Same for low values of k ($k \leq 4$).

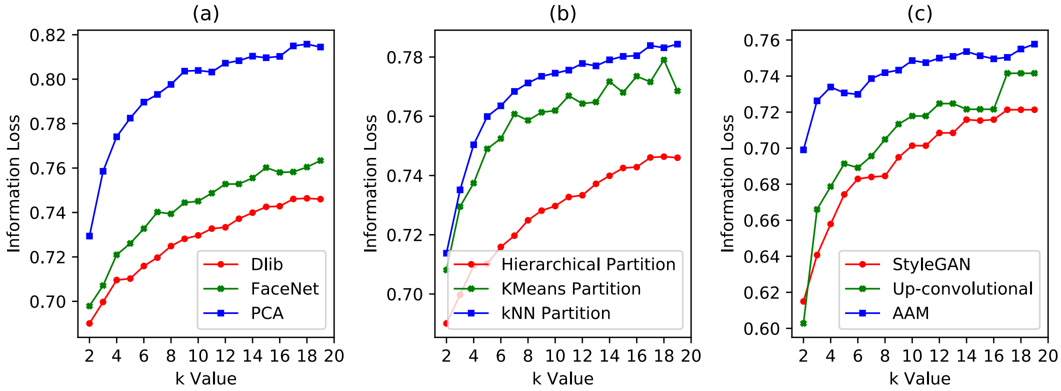

Figure 9: Information loss at different stages: (a) embeddings on CelebA, (b) clustering on CelebA, (c) generator on RafD

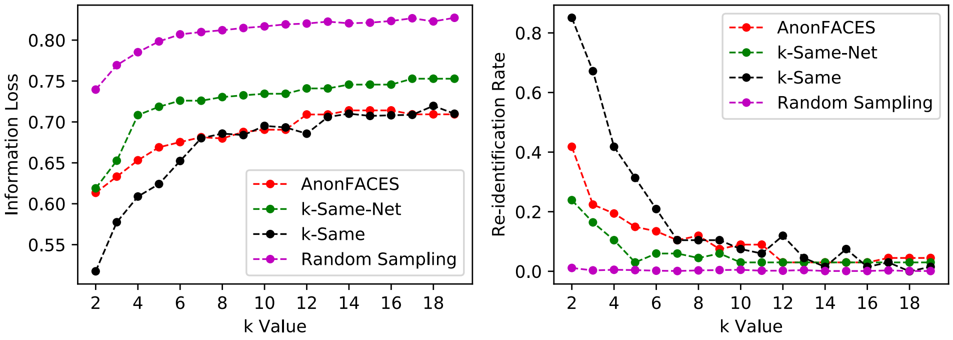

Figure 10: Comparison to the related works on the RafD dataset

4.4 Tuning: How to choose the k value?

AnonFACES provides several options to tune the privacy-utility trade-off. The first option is based on the information loss metric and the re-identification rate as demonstrated in the previous section. For example, in Figure 10, the plateaus between k equal to 12 and 20 or 5-10 result in negligible information loss and one can pick a higher k at little cost, and the overall curve gets flatter with higher k values as shown in Figure 11. These metrics enable a general analysis on the whole dataset, which mostly focuses on the loss of information and the risk of re-identification corresponding to each k value. Here, non-linear relations can be exploited.

With a small dataset such as RafD, which include few identities, we note that a visualization tool such as Dendrogram (see Figure 12) can also be used to help select a good k value. For example, by drawing a horizontal line across the Dendogram, we can find a place where the majority of data points (e.g. dashed-black horizontal line in Figure 12) has at least one cluster and a line where the minority of the data points has a cluster (e.g. dashed-magenta line in Figure 12). The lowest and the highest cluster sizes at the cuts can then provide approximate bounds for good k values. In the current example, at the lower cut, we have $k_{min} = 2$. This gives us clusters with members very close to each other; however, many outliers will need to be forced into their closest clusters. At the higher cut, we have $k_{max} = 5$. Here, the members are further away, but we have fewer outliers. The outliers are handled as described in Algorithm 1. The output after a single cut-tree is not the final output but it can give us a look on how the distribution of the clusters after cutting, hence an estimation for k value.

Depending on use cases and applications, the definition of utility can vary and the general metrics provided by us thus far are suboptimal compared to when the specific analysis is known and can inform which features to preserve. In those situations, we need to provide a list of attributes in advance and observe the change while the k value varies. One of the approaches we have experimented with AnonFACES is the homogeneity of attributes (note that with the help of the advancement in computer vision and deep learning, we have different tools for attributes extraction; this is out of scope of this paper). Considering a cluster C with k members, each member has a list of m-attributes $A = (a_1, \ldots, a_m)$. For each attribute $(a_i) \in A$ we calculate the entropy of that attribute for all k members in the cluster. A a_i-homogeneous cluster is one that has $Entropy_{a_i} = 0$. For each k value we find the rate of a_i-homogeneous clusters among all the clusters in the sampling dataset and observe the change while varying k. Figure 13 shows the example results for the age and gender attributes of the RafD dataset. As seen in the first two sub-figures, with $k = 2$, more than 80% of the clusters are homogeneous for both the attributes age and gender. With $k = 5$, the values rapidly reduce to around 70% for the case of age and to 60% for gender. Note that RafD is a very small dataset.

Similarly, we can define a $(a_i, \ldots, a_j)$-homogeneous cluster, where $(a_i, \ldots, a_j) \in A$, as the one that has $Entropy_{a_p} = 0$, for all $a_p \in (a_i, \ldots, a_j)$. As shown in the last subfigure in Figure 13, we still

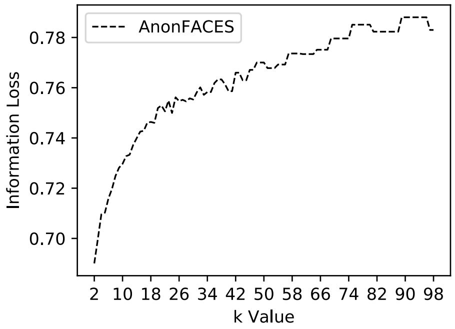

Figure 11: Experiment with maximum $k = 100$ on CelebA dataset

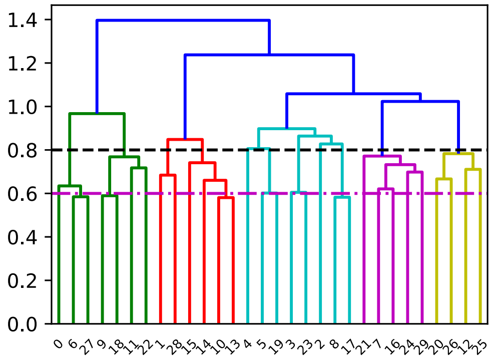

Figure 12: Dendrogram showing the hierarchical tree built from a sample of the RafD dataset: y-axis denotes the distance between clusters and x-axis denotes the index of data points in the dataset

have more than 70% of clusters that have both age and gender homogeneity with $k = 2$; however, the value falls below 50% for $k = 6$ before bounding back at around 60% for $k = 8$. Obviously, achieving multi-attribute homogeneity is more difficult comparing to single ones. In such a case where we need to preserve both attributes at a reasonable level ($\geq 60\%$), $k \leq 4$ is required.

5 SECURITY ANALYSIS

Newton *et al.* [24] consider an attacker that attempts to re-identify a de-identified face set by using face recognition software. They showed that traditional de-identification techniques such as pixelation, blurring, or masking some facial features are vulnerable to such an attacker. As with their original k-same algorithm, we need to show that the re-identification probability never exceeds $1/k$ to provide k-anonymity (Property S1). While this is trivially true (and independent of how sophisticated face recognition becomes) considering that all faces belonging to a cluster of size k are replaced by the same synthesized face, generated from the k faces belonging to the cluster, facial-recognition matches could still give the attacker additional knowledge or confirmations that could give them an advantage when combined with side information and, thus, resistance against facial recognition matches becomes necessary (Property S2).

In line with Newton *et al.* [24], we define three attacker types, depending on the images available to the attacker and what facial-recognition matches could tell them. We refer to the terms *gallery* and *probe* for the sets used during the attacks. The *gallery* is the set of known faces in the facial-recognition software, *i.e.,* $\mathcal{G} = \{\mathrm{G}_1, \ldots, \mathrm{G}_\mathrm{u}\}$, while the *probe* is the set of faces to recognize identities [35], *i.e.,* $\mathcal{P} = \{\mathrm{P}_1, \ldots, \mathrm{P}_\mathrm{v}\}$. An attacker always has access to face-recognition software as a re-identification tool. However, whether they have knowledge about the probe or gallery set depends on the specific attack scenario. We define re-identification as a match between an element of the probe and one or more elements of the gallery as returned by facial recognition.

In the naive recognition attack model, the attacker $\mathcal{A}_1$ tries to match original images to de-identified (synthesized) images by running the latter through face recognition but is not trying to interfere in the de-identification process. The original images (and any additional images), denoted by $\mathcal{I} = \{\mathrm{I}_1, \ldots, \mathrm{I}_\mathrm{s}\}$, are used as the *gallery* and the synthesized images, denoted by $\mathcal{D} = \{\mathrm{D}_1, \ldots, \mathrm{D}_\mathrm{t}\}$, as the *probe*. Therefore, in this model, $\mathcal{G} := \mathcal{I}$ and $\mathcal{P} := \mathcal{D}$.

In the reverse recognition attack model, the attacker $\mathcal{A}_2$ tries to match synthesized images, *i.e.*, the *gallery*, to original images. Thus, $\mathcal{P} := \mathcal{I}$ and $\mathcal{G} := \mathcal{D}$. The attacker has some additional knowledge: the original dataset. [2]

In the parrot recognition attack model, the attacker $\mathcal{A}_3$ attempts to match synthesized images to synthesized images. This experiment is performed by an attacker who can duplicate de-identification techniques and, therefore, is capable of de-identifying both *gallery* and *probe* images. This results in a comparison of de-identified images to de-identified images. Similar to the first attack model, the *gallery* might include more images than the ones being de-identified in the *probe*.

5.1 Evaluation of the Re-identification Rate

Rank-1 analysis. For evaluating the facial recognition resistance (S2) of AnonFACES, we conducted re-identification experiments on the de-identified images. In these experiments, our goal is to assess the risk of successful identification of a subject in the input set I based on the de-identified images in set D. We assume that the identities in input set I are known and using face recognition the task is to link the de-identified images in D to the identities in I. The re-identification risk is quantified based on recognition experiments on the person-specific CelebA [20] dataset as defined in 3.1. We randomly selected 50 identities from the CelebA dataset

[2] "By using the altered images as the gallery, the alterations due to de-identification may decompose and become dispersed through some number of principle components, thereby limiting the affects of the alterations when matching faces."[24]

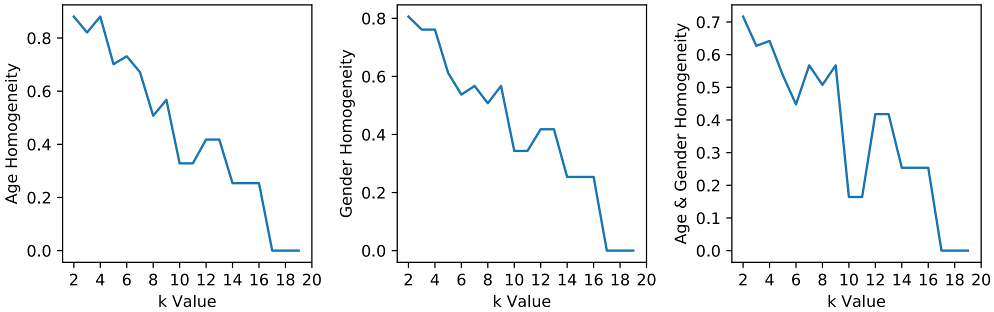

Figure 13: Rate of homogeneous cluster based on attributes age, gender, or both age and gender

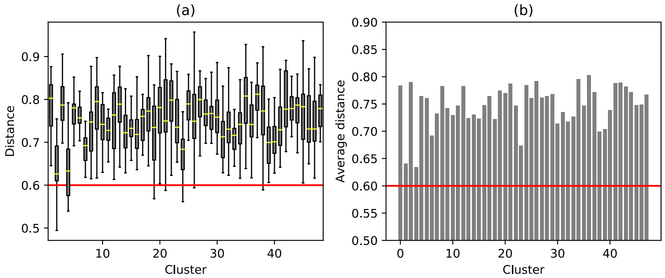

Figure 14: Pair-wise distance on CelebA, adjustment for re-identification risk limited to $1/k$: (a) boxplot of distances from original to synthesized faces, (b) average of pair-wise distance per cluster.

during each experimental run and constructed a probe set for de-identification which includes one image per identity. For the gallery set, we use 10,177 images which include one image for each identity in the probe set , different from the images used in the probe set. We then perform identification experiments using the Dlib [14] based python face recognition library and the constructed probe and gallery sets. To assess the effect of the k value, we repeat the process for k values 2, 4, and 8, when clustering the images of the probe set. We then present the performance as average rank one (Rank-1) recognition rates and the corresponding standard deviation for each k value over three experimental runs. The Rank-1 recognition reveals whether for a given probe image the top match in the gallery set identified by the face recognition is a correct match or not. In other words, Rank-1 recognition is true if the probe image and the top ranked image in the gallery set is of the same identity and false otherwise. Note that in our experiments, we know which identity was replaced by a given de-identified image. In an attack, any of the k identities in the same cluster would be a correct Rank-1 result, but with any k greater than 1, successful re-identification still does not tell an attacker whether the Rank-1 identity was in the original image or only in the cluster of size k that was used to synthesize the de-identified image.

Definition 5.1. The Rank-1 recognition rate (RR) for a probe set P can be formally defined as RR = $R1/|P|$, where $R1$ denotes the number of probe images in gallery set G that have been correctly recognized as top-ranked and $|P|$ is the size of the probe set.

Table 1 represents the Rank-1 recognition rate on the CelebA dataset before and after de-identification. While the average Rank-1 recognition rate before de-identification for the randomly selected identities of CelebA dataset is relatively low (around 78%) for this selection, it still provides a substantial contrast to the much lower Rank-1 recognition rates after de-identification. One possible reason for low recognition rate before de-identification could be due to the large gallery set and a very small probe set as pointed out in [13]. In the case of de-identified images, even with the k=2, the recognition rate reduces to 1.3%. This is well below the 2% achieved by k-same-Net [22] in similar experiments. As we increased the k value, the recognition rate decreased further. For example, with k=4 the average recognition rate is 0.67%, and with k=8 it is 0.0%. Thus, the overall experimental evaluation suggests that the risk of re-identification of images de-identified using AnonFACES is very low, especially if using a larger k value.

Table 1: Recognition performance of AnonFACES before and after de-identification over three repetitions of recognition experiments with images from CelebA dataset.

k value	Rank-1($\mu \pm \sigma$) Before deid.	Rank-1($\mu \pm \sigma$) After deid.
k=2	0.78667 ± 0.01155	0.01333 ± 0.01155
k=4		0.00667 ± 0.01155
k=8		0 ± 0

Naive and Reverse Recognition. Firstly, even though larger datasets can potentially lead to better de-identification results, we consider, for completeness, a dataset of the same size. In other words, the *gallery* only includes de-identified images. We evaluate attacks on both cluster level and database level:

- **Cluster Level Evaluation.** In Figure 14(a), we show a boxplot of an example where a sample of 1000 identities in CelebA dataset is used to cluster them into around 50 clusters so that each cluster has at least 20 members. In this figure, we can observe pair-wise distances from identities in clusters to their de-identified ones. We expect that the boxes are located in the middle of the boxplots and the median lines are in the middle of the boxes, which is the case for most of the clusters. Even though we still see some identities fall below the re-identification line (red line), the average pair-wise distances from the synthesized images to their corresponding cluster members (Figure 14(b)) are above the threshold for all clusters. In other words, cluster-wise, the matching from original images to synthesized ones or vice versa cannot be achieved, thwarting Attackers $\mathcal{A}_1$ and $\mathcal{A}_2$.

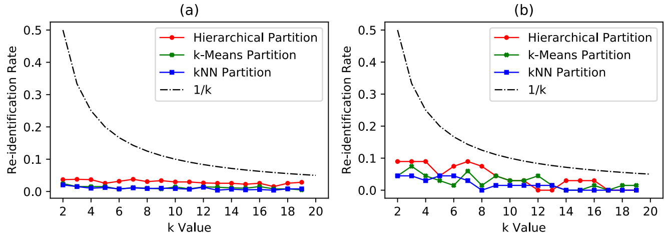

Figure 15: Comparison of re-identification rates between different clustering algorithms: (a) CelebA dataset, (b) RafD dataset

- **Database Level Evaluation.** Considering the whole database (CelebA and RafD), we see how the re-identification rates differ in Figures 15(a) and 15(b). As expected, the choice of clustering has an impact on the metric and it is more mixed for the case of RafD compared to CelebA. In these cases, we see that although kNN performs slightly better, all the lines are below the threshold of $1/k$. Thus, our technique is secure against both naive and reverse recognition tests with respect to k-anonymity (S1). To avoid any matches (S2), we adjust weights for identity mixing as described in Section 3.4.

Parrot Recognition. Among the three types of attacks, parrot recognition is the most advanced. With this type, attackers can have an advantage in breaking image de-identification algorithms, especially the traditional ones such as pixelation and blurring are at risk. The problem is that the existing approaches act as deterministic functions, meaning that with the same input, we always have the same result. By reverse engineering, making queries and observing the outputs, the attacker can perform this kind of attack even without the assumption that the de-identification algorithm is revealed.

Although StyleGAN provides a deterministic output, we can still introduce randomness to the identity-mixing process that provides the input. Since randomness can affect information loss, it has to be limited. We turn AnonFACES into a non-deterministic de-identification algorithm with a constraint on entropy.

Previously, we set weights in Equation (1): $w_i = 1/k$ for all $i \in [0, k]$, then change the weight according to re-identification rates in the Risk Assessment module. Setting w_i to random is equal to calculating LV_{mix} randomly, which is not what we want. Thus, we experiment with different setups of w_i to find a safety net for applying a random function on w_i. In this experiment, we are interested in the magnitude of w_i rather than individual values, so that we set $w_i = \alpha$ for $1 \leq i \leq k$ and $\alpha \in (0, 2)$, the range that StyleGAN still produces realistic results. We define our random function as selection of random values in uniform distribution of $(w_{mean} - \delta, w_{mean} + \delta)$, where δ is radius of the random range and w_{mean} is the mean of weight values.

Figure 16 shows the re-identification rate improvements achieved by both a basic weight adjustment in the Risk Assessment module (called "AnonFACES adjusted" in the figure) and when adding randomness to w_i ("AnonFACES rand") compared to a more basic implementation ("AnonFACES w/o adjustments"). Here, we show results for both (a) the information loss and (b) the re-identification probability when applied on the RafD dataset. The figure also shows clear trade-off between improved re-identification rates versus somewhat increased information loss that these privacy enhancements make.

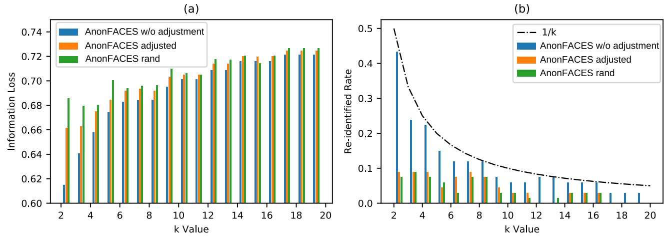

Figure 16: Effect of risk assessment and random weights on RafD: (a) information loss, (b) re-identification rate

6 CONCLUSION

In this paper, we presented AnonFACES, its methodology and system design, and compared its performance with other state-of-the-art solutions. At the core of the design is a novel methodology that helps us to, for the first time, quantify, improve, and tune the privacy-utility trade-off using an information loss metric. Our system evaluation demonstrates that the system meets the five design goals outlined in Section 3.1. In particular, the system achieved k-anonymity (S1) and facial recognition resistance (S2), while optimizing utility by minimizing information loss (E1), producing more natural looking de-identified images (E2), while also allowing for much greater flexibility in how the images are depicted, added, or removed (E3) than provided by prior work. By evaluating the system when applying different functions for each of the different modules with each of the three system components, we also provide insights into where some of the advantages arise for an example dataset. For example, it is shown that careful feature extraction (embedding) before applying the clustering, can significantly improve the clustering of similar images; an aspect mostly ignored in prior k-anonymity work. However, most importantly, the use of the quantifiable metrics that we introduce allows us to exploit any non-linear relations between privacy and utility to improve privacy or utility at limited loss of the other. We focused on a general notion of utility that does not prioritize the preservation of specific attributes over others and is thus suitable for the analysis-agnostic case. When it is known in advance which attributes are more important than others, the utility quantification can be adapted accordingly. Overall, the findings in this work demonstrate that solutions such as AnonFACES should allow the generation of highly privacy-preserving datasets without having to give up too much utility. This is important as it allows, for example, car manufacturers to effectively train their autonomous vehicles while complying with privacy laws.

Acknowledgements: This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

REFERENCES

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2StyleGAN: How to Embed Images into the StyleGAN Latent Space?. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 4432–4441.

[2] Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. Image2StyleGAN++: How to Edit the Embedded Images?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8296–8305.

[3] Elke Achtert, Hans-Peter Kriegel, and Arthur Zimek. 2008. ELKI: A software system for evaluation of subspace clustering algorithms. In *International Conference on Scientific and Statistical Database Management (SSDBM)*. Springer, 580–585.

[4] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. 2007. Efficient k-anonymization using clustering techniques. In *International Conference on Database Systems for Advanced Applications (DASFAA)*. Springer, 188–200.

[5] European Commission. 2018. European commission data protection rules. https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules/eu-data-protection-rules_en. (Accessed on 12/05/2019).

[6] Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. 2016. Learning to generate chairs, tables and cars with convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 4 (2016), 692–705.

[7] Oran Gafni, Lior Wolf, and Yaniv Taigman. 2019. Live face de-identification in video. In *IEEE International Conference on Computer Vision (ICCV)*. 9378–9387.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680.

[9] Ralph Gross, Edoardo Airoldi, Bradley Malin, and Latanya Sweeney. 2005. Integrating utility into face de-identification. In *International Workshop on Privacy Enhancing Technologies (PET)*. Springer, 227–242.

[10] Ralph Gross, Latanya Sweeney, Fernando De la Torre, and Simon Baker. 2006. Model-based face de-identification. In *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*. IEEE, 161–161.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[12] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4401–4410.

[13] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. 2016. The megaface benchmark: 1 million faces for recognition at scale. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4873–4882.

[14] Davis King. 2012. Dlib c++ library. http://dlib.net.

[15] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. 2010. Presentation and validation of the Radboud Faces Database. *Cognition and Emotion* 24, 8 (2010), 1377–1388.

[16] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-closeness: Privacy beyond k-anonymity and l-diversity. In *IEEE International Conference on Data Engineering (ICDE)*. IEEE, 106–115.

[17] Tao Li and Lei Lin. 2019. AnonymousNet: Natural face de-identification with measurable privacy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.

[18] Yuezun Li and Siwei Lyu. 2019. De-identification without losing faces. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*. 83–88.

[19] Jun-Lin Lin and Meng-Cheng Wei. 2008. An efficient clustering method for k-anonymization. In *Proceedings of the International Workshop on Privacy and Anonymity in Information Society (PAIS)*. 46–50.

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2018. Large-scale celebfaces attributes (CelebA) dataset. *Retrieved August* 15 (2018), 2018.

[21] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. 2006. l-diversity: Privacy beyond k-anonymity. In *International Conference on Data Engineering (ICDE)*. IEEE, 24–24.

[22] Blaž Meden, Žiga Emeršič, Vitomir Štruc, and Peter Peer. 2018. K-same-Net: K-anonymity with generative deep neural networks for face deidentification. *Entropy* 20, 1 (2018), 60.

[23] Lily Meng, Zongji Sun, Aladdin Ariyaeeinia, and Ken L Bennett. 2014. Retaining expressions on de-identified faces. In *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 1252–1257.

[24] Elaine M Newton, Latanya Sweeney, and Bradley Malin. 2005. Preserving privacy by de-identifying face images. *IEEE Transactions on Knowledge and Data Engineering* 17, 2 (2005), 232–243.

[25] NVIDIA. [n.d.]. StyleGAN - Official TensorFlow Implementation. https://github.com/NVlabs/stylegan. (Accessed on 04/15/2020).

[26] Yi-Lun Pan, Min-Jhih Haung, Kuo-Teng Ding, Ja-Ling Wu, and Jyh-Shing Jang. 2019. k-Same-Siamese-GAN: k-Same Algorithm with Generative Adversarial Network for Facial Image De-identification with Hyperparameter Tuning and Mixed Precision Training. *arXiv preprint arXiv:1904.00816* (2019).

[27] Slobodan Ribaric, Aladdin Ariyaeeinia, and Nikola Pavesic. 2016. De-identification for privacy protection in multimedia content: A survey. *Signal Processing: Image Communication* 47 (2016), 131–151.

[28] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65.

[29] Branko Samarzija and Slobodan Ribaric. 2014. An approach to the de-identification of faces in different poses. In *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 1246–1251.

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 815–823.

[31] Zongji Sun, Li Meng, and Aladdin Ariyaeeinia. 2015. Distinguishable de-identified faces. In *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Vol. 4. IEEE, 1–6.

[32] Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 557–570.

[33] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[34] Yifan Wu, Fan Yang, Yong Xu, and Haibin Ling. 2019. Privacy-Protective-GAN for privacy preserving face de-identification. *Journal of Computer Science and Technology* 34, 1 (2019), 47–60.

[35] Shaohua Kevin Zhou and Rama Chellappa. 2005. Face Recognition from Still Images and Videos. In *Handbook of Image and Video Processing (Second Edition)*, Alan C. Bovik (Ed.). Academic Press, Burlington, 1235 – 1250.

Appendix A ADDITIONAL COMPARISONS OF VISUAL RESULTS WITH RELATED WORKS

In this appendix we compare visual results of AnonFACES with recent results from [18] and [7]. As demonstrated in Figure 17, the results in [18] have a blurring effect which reduces the image quality while some identity features (e.g. general facial landmark) are still similar. AnonFACES clearly outperforms [18] in terms of naturalness and has a noticeably better anonymizing effect. In Figure 18 the results from [7] are shown to have minimal difference in terms of identity features such as general facial landmark features, eyes, and mouth area. AnonFACES again provides visibly better anonymization. In both cases, our results still preserve non-identity information such as emotion, age, gender, and skin color. Note, however, that [18] and [7] only replace some (identifying) facial features, presumably to work with videos, whereas AnonFACES replaces the entire image. Restricting the area to within the face would allow for a more direct comparison but is outside of the scope of this paper.

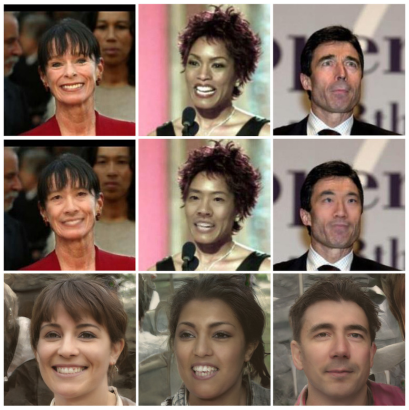

Figure 17: Comparison with Li *et al.* [18]: first row: input images, second row: results from [18], third row: AnonFACES results using k=2

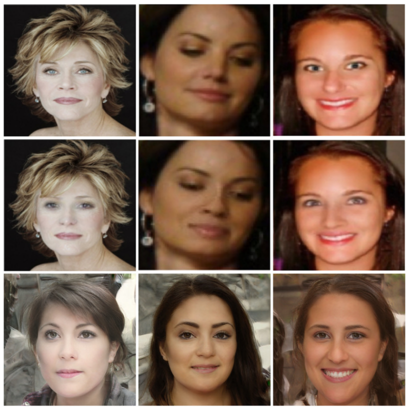

Figure 18: Comparison with Gafni *et al.* [7]: first row: input images, second row: results from [7], third row: AnonFACES results using k=2