Deep Autoencoders for Attribute Preserving Face De-identification

Paraskevi Nousi*, Sotirios Papadopoulos, Anastasios Tefas, Ioannis Pitas

Department of Informatics, Aristotle University of Thessaloniki, Greece

Abstract

The mass availability of mobile devices equipped with cameras has lead to increased public privacy concerns in recent years. Face de-identification is a necessary first step towards anonymity preservation, and can be trivially solved by blurring or concealing detected faces. However, such naive privacy protection methods are both ineffective and unsatisfying, producing a visually unpleasant result. In this paper, we tackle face de-identification using Deep Autoencoders, by finetuning the encoder to perform face de-identification. We present various methods to finetune the encoder in both a supervised and unsupervised fashion to preserve facial attributes, while generating new faces which are both visually and quantitatively different from the original ones. Furthermore, we quantify the realism and naturalness of the resulting faces by introducing a diversity metric to measure the distinctiveness of the new faces. Experimental results show that the proposed methods can generate new faces with different person identity labels, while maintaining the facelike nature and diversity of the input face images.

Keywords: Face De-identification, Attribute Preservation, Deep Learning, Autoencoders

*Corresponding author

Email addresses: paranous@csd.auth.gr (Paraskevi Nousi), pasotirio@csd.auth.gr (Sotirios Papadopoulos), tefas@csd.auth.gr (Anastasios Tefas), pitas@csd.auth.gr (Ioannis Pitas)

1. Introduction

In modern times, the mass availability of mobile devices equipped with cameras, such as mobile phones, camera vans used for city mapping purposes or dash cameras, and more recently, drones, has lead to increased public privacy concerns. As unknowingly photographed individuals often wish to maintain their anonymity, companies which manage databases of such images opt for de-identification methods to provide this anonymity. This is now enforced, at least in the European Union, by the recent GDPR legislation [1]. Therefore, a good practice would be to install privacy by design systems as close as possible to surveillance cameras, and even cameras used for entertainment purposes, such as on drones, to ensure privacy protection.

As a person's face is amongst the most significant biometric features when it comes to person identification, both by humans [2], and by computers [3], typically face de-identification suffices for anonymity preservation. A standard de-identification method comprises of face detection as a first step and blurring of the detected facial regions to achieve de-identification. As an example, in the images shot by a Google Camera Car for mapping purposes, when a face is detected, the face is blurred to hinder the recognition of that person both by computers and humans alike. Although unimportant in this context, such blurring processes produce a visually unpleasing result. Furthermore, it has been shown that such naive techniques can be defeated, for example via parrot recognition [4]. In addition, recent studies show that pixelation and blurring do not ensure proper de-identification [5, 6], as in the absence of clear facial features, humans will look for contextual clues to recognize a person, with surprising success. Thus, more advanced face de-identification methods, in terms of effectiveness and utility of the resulting images, must be investigated.

With the recent advances of Machine Learning (ML) in face detection and recognition, face de-identification methods can become much more effective and efficient. Face detection and recognition accuracy has been greatly improved [7, 8]. Thus, de-identification methods can be better evaluated using learned face verification and recognition models. Furthermore, the aesthetic quality of the de-identified images can be improved with Machine Learning, which offers more sophisticated solutions than face blurring.

In this paper, we exploit Autoencoders (AEs), extracting meaningful low-dimensional feature representations of facial images, to tackle the problem of face de-identification. Interestingly, we find that the information loss caused

Figure 1: Before (left) and after (right) application of the proposed de-identification method. The identity of the subject is altered in a manner which preservers basic visual attributes, such as gender, pose, emotion and age, thus maintaining the naturalness of the image.

by the dimensionality reduction that AEs perform, yields facial images which, while visually similar to the original ones, are recognized as different by state-of-the-art face recognition systems. We exploit this inherent ability for face de-identification, through the generation of facial images with different identity labels than the original one, both visually and quantitatively.

The main contribution of this paper lies in finetuning the encoding part of a standard autoencoder to perform de-identification in the latent space. After finetuning the encoder, the network can then perform face de-identification in an end-to-end fashion, by forward passing the facial image through the modified encoder, which changes the identity of the face while preserving other attributes, and subsequently the decoder, which then reconstructs a new face. Extensive experimental results using both supervised and unsupervised learning for the encoder finetuning step indicate the effectiveness of the proposed method. Figure 1 demonstrates the ability of the proposed method to alter the identity of a subject effectively, while maintaining the naturalness of the image by creating a new face which inherently blends in with the context of the original face. Finally, a *diversity* measure is introduced, to quantify the quality of the generated faces, in terms of realism and utility of a de-identified image.

The rest of this paper is organized as follows. Section 2 contains a state of the art review and theoretical comparisons to this work. In Section 3 the tools used in the proposed method are introduced, and the method is described in detail. Section 4 presents the results of our experimental study, while the conclusions drawn are summarized in Section 5.

2. Related Work

Recent methods used for face de-identification focus on finding the balance between preserving global facial attributes and simultaneously hindering face recognition by altering the more refined facial attributes. The visual quality of the de-identified image is addressed by using models which generate new facial images of high quality. An overview of various face de-identification methods can be found in [9]. Despite recent interest, there are no widely accepted benchmarks for face de-identification, and various face recognition datasets are used in conjunction with state-of-the-art face recognition systems to evaluate de-identification systems.

2.1. De-identification and k-Anonymity Theory

The k-Anonymity theory [10] imposes that in a set of de-identified people, each person cannot be distinguished from at least $k-1$ other people in the same set. In [4], the k-Same algorithmic family was introduced, providing a de-identification method with theoretical guarantees of anonymization. The family was later enriched with the addition of the model-based k-Same-M method [11], which is more robust to alignment and other variation issues in a gallery set. Based on the k-Same-M family, in [12], multiple facial attribute classifiers as well as a face verifier are used in conjunction with a face image synthesis model, relying on heavily annotated datasets with multiple facial attribute features. The attribute classifiers are then used to measure the attribute similarity between a test sample and other samples from the gallery set. Compared against the k-Same-M method, the proposed method greatly reduces face recognition rates. In [13], the concepts of k-anonymity, l-diversity and t-closeness are directly tied to a proposed neural architecture, for effective anonymization of facial images.

2.2. Neural Networks & De-identification

Many tasks related to face analysis, such as face detection, recognition and verification [8, 7], have benefited from the recent success of Deep Learning methods in the Computer Vision field. Convolutional neural networks revolutionized the way many face-related problems are handled, from face recognition [14], facial pose recognition in the wild [15], to attribute alteration [16]. Neural networks were used in [17], in combination with a background subtraction and segmentation scheme, for pedestrian face de-identification in

video. A style transfer algorithm is applied partially to change the appearance of a face detected in an image, after segmentation has revealed salient areas (i.e., people). The result is a de-identified image whose visual qualities depend heavily on the reference style image. However, it can be argued that changing the artistic style of a detected face doesn't effectively hinder recognition, while providing a somewhat unrealistic result.

In [18], a Generative Adversarial Network (GAN) is used for inpainting facial details after facial landmarks have been identified. This preserves the pose of the original face, while constructing a facial image of high quality which is visually pleasing and more effective than blurring methods for de-identification purposes. Generative networks were also used in [19], to generate artificial faces which are then blended to create a new identity. Using a deep VGG-like network, feature vectors are extracted from a gallery set, and the k-Nearest Neighbors of each sample in the set are computed. These are then fed into the generative network which generates the new identity. The generated face is blended into the context of the original one in a separate step.

More recently, GANs were used for full body as well as face de-identification [20], addressing the problem of identity recognition via non physiological characteristics. The networks are trained to produce new samples of clothing as well as faces which match the probability distribution of the training dataset. Then, during deployment, new clothing items and faces are generated and inpainted in place of the original ones, while attempting to protect the *naturalness* of the image against artifacts inserted by the GAN. GANs have also been used for style transfer purposes, which is closely related to de-identification, in [16]. Finally, adversarial training was used in [21], to train a generative network to produce new faces while preserving the action that the original person was performing. A deep resisual-based architecture was used with impressive results. However, despite the action preservation, no other attribute preservation occurs with regards to the faces themselves (e.g., gender, age or emotion).

Autoencoders & This Work. In contrast to the above methods, in this work Autoencoders [22] are used, and it is shown experimentally that they are tools capable of achieving good de-identification performance, while producing high quality reconstructions. Deep autoencoders have recently been proposed to solve tasks such as one-shot face recognition [23], unconstrained face recognition [24, 25], face alignment [26] and face hallucination [27] among

others. The features extracted by deep AEs have been shown time and time again to be significantly useful to such tasks. Adversarially trained autoencoders were proposed in [28] for image generation and manipulation, and stacked Wasserstein autoencoders were proposed in [29] and applied to facial images with interesting results.

Because of the information loss that occurs when extracting representations of lower dimensionality, we show that even standard AEs reconstruct faces with altered identities. With an encoder finetuned for the purpose of de-identification, the de-identification performance increases even further and the produced images are also visually and quantitatively dissimilar to the original ones. The naturalness of the resulting images is quantified and measured via faceness and diversity metrics, highlighting the ability of the proposed autoencoders to produce images which highly resemble diverse faces, with different identities than the original ones. Furthermore, the proposed method is very lightweight and capable of processing hundreds of faces at very high speeds, as it does not require online training or any additional processing, such as image segmentation. Thus, in conjunction with a fast face detector [30], a real-time de-identification system based on the proposed method can be deployed, even on mobile devices.

3. Proposed Method

3.1. Autoencoders

Autoencoders (AEs) are neural networks which map their input data to itself, through multiple levels of non-linear neurons [22, 31, 32]. Thus, the input and output layers consist of as many neurons as is the dimension of the data. Such networks are comprised of an encoding part, which maps the input to an intermediate representation, and a decoding part, which maps the learned intermediate representation to the desired output, and is layer-wise symmetrical to the encoding part.

Typically, an AE is used for dimensionality reduction as well as feature extraction, which means that the intermediate representation learned is of lower dimension than the input data. The layers of both parts of the network ($l = 1, \ldots, l_{enc}, \ldots, L$, where l_{enc} is the encoding layer), are accompanied by weights $\mathbf{A}^{(l)} \in \mathbb{R}^{D_l \times D_{l-1}}$ which multiply each layer's input to produce an output, where D_l is the dimension the output at layer l. A bias term $\mathbf{b}^{(l)} \in \mathbb{R}^{D_l}$ is also added to the output of the neuron, and a non-linearity

$s(\cdot)$ called the activation function of the neuron is applied to this output to produce the neuron's activation value.

In the context of fully connected layers, the output $\mathbf{x}_{out}^{(l)} \in \mathbb{R}^{D_l}$ of the l-th layer is given by:

$$\mathbf{x}_{out}^{(l)} = s(\mathbf{A}^{(l)}\mathbf{x}_{in}^{(l)} + \mathbf{b}^{(l)}) \tag{1}$$

where $\mathbf{x}_{in}^{(l)}$ is the input to the l-th layer, which is equal to the output of the previous layer, or:

$$\mathbf{x}_{in}^{(l)} = \begin{cases} \mathbf{x} & , \quad l = 0 \\ \mathbf{x}_{out}^{(l-1)}, & \quad l > 0 \end{cases} \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^{D_0}$ denotes an input sample, D_0 being the dimensionality of the input.

For convolutional layers, the network's input is a three-dimensional tensor representing an image as its pixel intensities, i.e., $\mathbf{x} \in \mathbb{R}^{C_{in}^{(0)} \times H \times W}$ where H, W are the image's height and width and $C_{in}^{(0)}$ is the number of channels of the input image (e.g., 3 in RBG images). In this setting, each layer is parameterized by $C_{out}^{(l)}$ filters $\mathbf{f} \in \mathbb{R}^{C_{out}^{(l)} \times n \times n \times C_{in}^{(l)}}$, and $C_{out}^{(l)}$ biases $b_c^{(l)} \in \mathbb{R}$. The output of the l-th layer is computed as:

$$\mathbf{x}_{out,c}^{(l)} = b_c^{(l)} + \sum_{k=0}^{C_{in}^{(l)}-1} \mathbf{f}_{c,k}^{(l)} * * \mathbf{x}_{in,k}^{(l)} \tag{3}$$

where $\mathbf{x}_{out,c}^{(l)} \in \mathbb{R}^{H_l \times W_l}$ is the c-th channel of the output tensor with $c \in \{0, 1, \ldots, C_{out}^{(l)}\}$, $b_c^{(l)}$ is the c-th bias value, $\mathbf{f}_{c,k}^{(l)} \in \mathbb{R}^{n \times n}$ is the k-th channel of the c-th filter, $\mathbf{x}_{in,k}^{(l)} \in \mathbb{R}^{H_{l-1} \times W_{l-1}}$ is the k-th channel of the input tensor and finally the double star symbol, $**$, denotes the 2D convolution function.

As in standard convolution, the input image must be padded spatially in accordance with the size of the convolution filter to preserve its dimensions. When no padding is added, the dimensions of the resulting volume are smaller than the original. To reduce the dimension faster, strided convolutional or pooling layers may be used. As an example, a max pooling operation with a 2×2 kernel and stride 2 will reduce the input volume by a factor of two immediately, by choosing the maximum response out of each 2×2 non-overlapping input region, due to the stride. Although this causes some information loss, it can be compensated for by increasing the number of filters in the convolutional layers, to assure sufficient information is encoded for the reconstruction process.

The network parameters, i.e., the weights or filters and biases, can be learned using the backpropagation algorithm [33], in combination with an optimization method, such as Stochastic Gradient Descent (SGD) [34], to optimize an objective function. For autoencoders, the objective is typically to minimize the reconstruction error, i.e., difference of the output and the input, in terms of a differentiable function, such as the squared l_2-norm:

$$\ell(\mathbf{x}_i, \mathbf{x}_{i,out}^{(L)}) = \|\mathbf{x}_i - \mathbf{x}_{i,out}^{(L)}\|_2^2 \tag{4}$$

The objective of the network is to minimize the mean of errors over all data samples:

$$\mathcal{L}(\mathbf{X}, \mathbf{X}_{out}^{(L)}) = \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \ell(\mathbf{x}_i, \mathbf{x}_{i,out}^{(L)}) \tag{5}$$

where $\mathbf{X}, \mathbf{X}_{out}^{(L)}$ are the matrices containing all input samples $\mathbf{x}_i$ and reconstructed samples $\mathbf{x}_{i,out}^{(L)}$, for $i = 1, \ldots, N$, and θ is the set of the network's optimizable parameters, i.e., $\theta = \{\mathbf{A}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L}$ for networks comprised of fully connected layers, or $\theta = \{\mathbf{f}^{(l)}, \mathbf{b}^{(l)}\}_{l=1}^{L}$ for convolutional networks.

As the network's output is a function of its input and the network's weights and biases, the optimization of this loss function is tied to updating the parameters of the network towards the minimization of the reconstruction error. The process of updating the network's parameters is referred to as the network's training process. As the training process converges, the activations of the intermediate layers can be used as representations of the input data for other tasks, such as classification and clustering. Let $\mathbf{x}_{enc}$ denote the output of the l_{enc} layer, i.e., $\mathbf{x}_{enc} = \mathbf{x}_{out}^{(l_{enc})} \in \mathbb{R}^d, d < D$. Then $\mathbf{x}_{enc}$ can be used as the low-dimensional representation of the data in the subsequent classification task.

It's worth noting that the described process is fully unsupervised, meaning that the label information of the data is not utilized during the training process. However, when the label information is available, its utilization in the training process of an AE could intuitively improve the quality of the features produced [35].

3.2. Autoencoders for Face De-identification

Let $\mathbf{x}_i$ represent a facial image and $\mathbf{x}_{i,enc}$ represent its encoded feature vector, learned by a standard Autoencoder. Due to the dimensionality reduction, the encoded feature vector is a lossy compressed version of the original image. As a direct result, the reconstruction will not be perfect even

though the network has been trained to minimize the error between the reconstruction and the input. In fact, this loss of information is enough to hinder an artificial face recognition system's accuracy, despite the fact that the reconstructed image may be visually very close to the original, much like adversarial examples designed for the sole purpose of confusing neural networks [36]. This important AE property is further demonstrated in Section 4.

Based on this observation, and the fact that the encoded representation contains information more useful than pixel intensities, we disintegrate the autoencoder into its encoder and decoder parts and focus on finetuning the encoder to produce representations which visibly alter the identity of its input. To achieve de-identification for a given sample $\mathbf{x}_i$, a target representation $\mathbf{z}_i$ is defined, and the encoder is trained to learn this representation by minimizing the following loss function over all samples:

$$\ell_{deid}(\mathbf{x}_{i,enc}, \mathbf{z}_i) = \|\mathbf{x}_{i,enc} - \mathbf{z}_i\|_2^2. \tag{6}$$

The choice of targets $\mathbf{z}_i$ is detailed in the following subsections.

Supervised Attribute Preserving De-identification. When attribute labels are available, i.e., each sample $\mathbf{x}_i$ is accompanied by a set $\mathcal{A}_i$ of attributes such as identity, gender, ethnicity, pose, facial expression etc., these can be exploited to produce more visually pleasing de-identified faces. This can be achieved by forcing the encoded representation to lie away from the representation vectors with conflicting attributes and closer to representations with desirable attributes. We denote attributes as desirable or undesirable in the context of face de-identification. For example, identity is an undesirable attribute, and the representations should be shifted away from samples with the same identity. Other attributes, such as gender, pose and expression, are desirable for attribute preservation, so that the de-identified face will naturally blend into the context of the original face.

Formally, we define the target representation of the i-th sample, $\mathbf{y}_i$, as:

$$\mathbf{y}_i = (1 - \alpha)\mathbf{x}_{i,enc} + \alpha \mathbf{P}_i \mathbf{X}_{enc} \tag{7}$$

where $\mathbf{P} \in [0, 1]^{N \times N}$ is an *attraction* matrix, whose rows describe the effect of each sample in the set of encoded feature vectors $\mathbf{X}_{enc} \in \mathbb{R}^{N \times D_{enc}}$ on the i-th sample, i.e., the i-th row $\mathbf{P}_i \in [0, 1]^N$ contains weights by which all samples are linearly combined, to form the new target. The hyperparameter α

weighs the effect of the shift on the original sample, i.e., small values maintain more of the original features. For convolutional networks, the targets can be extracted either by reshaping the encoded feature representations as vector and subsequently reshaping them into tensors of the same size as the encoded tensor, or by extracting a target per each grid cell of the encoded representation.

To ensure attribute preservation, the weight P_{ij} linking sample $\mathbf{x}_j$ to sample $\mathbf{x}_i$ must be large if both samples are characterized by the same desirable attribute, or close to zero if they have undesirable to retain (clashing) attributes such as the same identity, or different poses. Let us denote by $\mathcal{D}_i$ the set of samples with the same desirable attributes as sample $\mathbf{x}_i$. Then, the weights P_{ij} can be formulated as:

$$P_{ij} = \begin{cases} \frac{1}{|\mathcal{D}_i|}, & \text{if } \mathbf{x}_j \in \mathcal{D}_i \\ 0 \quad , & \text{otherwise} \end{cases} , \tag{8}$$

indicating that each sample in the set of samples with desirable attributes $\mathcal{D}_i$ contributes equally to the target feature $\mathbf{y}_i$, with force equal to the inverse of the set's cardinality.

For large datasets, the set of samples with desirable attributes can be constricted to only include the nearest neighbors of each sample. Denoting this set by $\mathcal{D}_i^n$, the weights $\mathbf{P}_i$ can be extracted by Equation (8) by simply replacing $\mathcal{D}_i$ by $\mathcal{D}_i^n$. Thus, more complex weights can be produced by defining other sets of samples, such as for example, the nearest neighbors of each will also lie closer to the mean vector of all samples, or to the center of the same cluster as this sample.

Although the above process preserves the desirable attributes, it does not by itself guarantee that the identity of the person will be changed. Thus, the target extraction process can be enhanced by defining a set of samples with clashing attributes, $\mathcal{C}_i$, for each sample, such that the sample is moved away from the samples in this set. Formally, the new targets are defined as:

$$\mathbf{z}_i = (1 + \beta)\mathbf{y}_i - \beta \mathbf{Q}_i \mathbf{Y} \tag{9}$$

where the matrix $\mathbf{Q} \in [0, 1]^{N \times N}$ defines the relationships to be suppressed in the resulting representation, acting on the targets $\mathbf{Y} \in \mathbb{R}^{N \times D_{enc}}$ acquired by Equation (7). The *repulsion* matrix $\mathbf{Q}$ can be defined similarly to the attraction matrix $\mathbf{P}$ from Equation (8), by defining the proper sets of samples with conflicting attributes. Let $\mathcal{C}_i$ denote the set of samples with attributes

clashing those of the i-th sample. Such attributes include the same identity, and different properties such as the pose, gender and expression of the depicted people. After selecting this set, the repulsion weights for each sample are formulated as:

$$Q_{ij} = \begin{cases} \frac{1}{|\mathcal{C}_i|}, & \text{if } \mathbf{x}_j \in \mathcal{C}_i \\ 0 \quad , & \text{otherwise} \end{cases} , \tag{10}$$

where values close to one indicate strong repulsion, and values close to zero indicate little to no repulsion at all. Intuitively, if a sample lies far away from the i-th sample, it should not contribute to its target representation, whereas samples lying close to it should have a more significant effect on it. Thus, we can define a set of neighboring clashing samples as $\mathcal{C}_i^n$ for sample $\mathbf{x}_i$ and encode this information into the target representation by replacing $\mathcal{C}_i$ in Equation (10) with $\mathcal{C}_i^n$.

By combining the attraction and repulsion matrices, the extracted target representations force the encoder to learn an attribute-preserving de-identification function, whose output is fed to the decoder during the test phase and produces a new face that is both visually different and not recognizable as the same person by state-of-the-art face recognition systems, as will be demonstrated in Section 4.

Unsupervised De-identification. As the process of labeling data is very expensive, and unlabeled data exist in surplus, ideally Machine Learning methods should be capable of exploiting this vast volume of unlabeled data and still producing meaningful representations and results. Although labels are not required during the test phase of the proposed method, as described so far, they are still required in abundance for the training phase.

To avoid this costly process and enable our method to work in a purely unsupervised manner, we exploit the autoencoder's ability to uncover meaningful low-dimensional representations, in the sense that samples which lie close to each other in the latent subspace will have similar attributes whereas samples lying further away from each other will have dissimilar attributes. Autoencoders have been shown to be able to group samples by their attributes, as demonstrated in recent works on clustering which use autoencoders as a feature extractor [37, 38].

Based on this observation as well as the fact that the reconstructed facial images act as adversarial examples confusing state-of-the-art face recognition

systems, we formulate attraction matrices without explicit attribute information, which force the encoded representations to be decoded into visibly and quantitatively different identities. For each sample $\mathbf{x}_i$, we define a set $\mathcal{D}^{\mathcal{U}}{}_i$ of samples which are more probable to be characterized by the same set of attributes, for example its n closest neighbors, so as to preserve those in the reconstruction phase. As distances are also more meaningful in the low-dimensional space [39], neighbors found by measuring distances at the latent space are more likely to be semantically related to each other. In general, there will also be a set $\mathcal{C}^{\mathcal{U}}{}_i$ of samples with attributes conflicting with those of sample $\mathbf{x}_i$.

It is thus intuitive to first cluster the faces in the low-dimensional space, using a clustering algorithm such as k-means, by considering the resulting clusters as groups of samples with the same attributes. As aforementioned, autoencoders are capable of producing representations which can then be grouped into semantically similar clusters. Thus, the assumption that samples belonging to the same cluster will have the same or similar attributes, is not a far-fetched one, and one could define $\mathcal{D}^{\mathcal{U}}{}_i$ as the set of samples in the same cluster as $\mathbf{x}_i$. Respectively, the set of samples belonging to different clusters can be assumed to have different attributes, and thus constitute the set of repulsion samples $\mathcal{C}^{\mathcal{U}}{}_i$. Having chosen the attraction and repulsion sets, the new targets can be acquired in the same way as in the supervised paradigm, i.e., by combining Equations (7) and (9) using these sets.

However, this set of attributes may also include the identity of the depicted person, which clashes with the purpose of de-identification. To combat this, we first compute mean representations corresponding to either an entire face or different facial parts, e.g., left and right eyes, nose, mouth etc., by extracting feature vectors from the corresponding grid cells of the encoded representations. We then restrict the samples contributing to the attraction matrix $\mathbf{P}$, by choosing only one of the nearest neighbors to contribute to the shift. This leads to a more crisp target, whose identity is different than the original, but whose facial features are structurally close to the original ones. As an example, this neighbor can be chosen to be each sample's n-th closest neighbor or the one lying the closest to the mean face or facial part out of this set, which we denote by $\mathbf{n}_i$. The latter process ensures attribute preservation, given the autoencoder produces semantically meaningful representations, while attempting to modify the identity of the depicted person by moving its latent representation towards one that is closest to the mean representation, whose identity is a mixture of all identities in the dataset,

i.e., it has been anonymized because it resembles all faces equally, and thus none of the faces more than any other at the same time.

Formally, the target extraction can be summarized as:

$$\mathbf{z}_i = (1 - \alpha)\mathbf{x}_{i,enc} + \alpha\mathbf{n}_i. \tag{11}$$

In this unsupervised context, finding the balance between de-identification and attribute preservation depends on the number of neighbors chosen to fill the set $\mathcal{D}^{\mathcal{U}}{}_i$ with.

Figure 2 illustrates the effect of some of the aforementioned shift types, for thirty samples of the LFW dataset [40], corresponding to ten facial images from three different subjects, denoted by different shapes. Each subject is also characterized by an additional attribute, either "smiling" or not, denoted by different colors. Figure 2a, shows the original samples encoded by a standard convolutional autoencoder and projected into 2D space via PCA [41]. Although samples with similar attributes lie close together, no prominent clusters of same-attribute samples appear. In Figure 2b, the samples are shifted towards the mean of samples with the "smiling" attribute as well as away from the mean of samples without the attribute. The resulting representations are well separated in terms of the attribute in question.

In Figure 2c, the original samples are shifted towards their nearest same-attribute neighbors and away from their closest neighbors without the attribute. The resulting representations are well separated in terms of the attribute, and the identities are entangled, which is a desirable side-effect. Finally, in Figure 2d, the samples are first clustered and then shifted towards their cluster centers. Although the separation is unsupervised, the formed clusters are quite uniform in terms of their attributes.

4. Experimental Study

4.1. Performance Measures

Faceness. We measure the faceness of a de-identified facial image by its ability to be confidently detected as a face by a neural network based face detector, provided by dlib [42]. Formally, for a set of N de-identified faces $\tilde{\mathbf{X}}$:

$$FCNS(\tilde{\mathbf{X}}) = \frac{1}{N}\sum_{i=1}^{N} f_{det}(\tilde{\mathbf{x}}_i) \tag{12}$$

(a) (b) (c) (d)

Figure 2: Encoded samples from the LFW dataset corresponding to male and female subjects (colors), with "youth" and "white" attributes (squares and circles respectively). (a) Original encoded samples, projected into 2D space by PCA [41]. (b) Samples shifted towards the center of their attribute and away from the center of samples without the attribute. (c) Samples shifted towards their closest same-attribute neighbors and away from their closest neighbors without the attribute. (d) Samples shifted towards their cluster center, uncovered by k-Means.

where f_{det} returns 1 if the de-identified face is detected as a face and 0 otherwise.

De-identification. We measure the ability of the network to produce faces with different identities from the original samples by measuring the de-identified face's similarity to other images of the depicted person. For this purpose, we use pretrained face recognition systems provided by dlib. Formally, for a set of de-identified facial images $\tilde{\mathbf{X}}$ and their original counterparts $\mathbf{X}$:

$$DEID(\tilde{\mathbf{X}}, \mathbf{X}) = \frac{1}{N} \sum_{i=1}^{N} 1 - f_{rec}(\tilde{\mathbf{x}}_i, \mathbf{x}_i) \tag{13}$$

where the $f_{rec}(\cdot, \cdot)$, given two input facial images, returns 1 if they correspond to the same person and 0 otherwise. Internally, a face recognition system from dlib is used to extract feature vectors corresponding to the two input faces. Then, the euclidean between the features is measured and thresholded to produce the binary output. In our experiments we set this threshold to 0.6, adopted from the official dlib implementation[1], which achieves an accuracy of 99.38% on the LFW benchmark.

Although the final goal is of the proposed system for the person to be non-identifiable by both humans and computers, we believe the DEID metric

[1] http://dlib.net/face_recognition.py.html

is a valid measure of de-identification especially in combination with the FCNS metric, based on the increasingly successful performance of modern face recognition systems.

Diversity. We also measure the ability of the de-identification model to produce diverse faces, by measuring their similarity using the face recognition system used for the $DEID$ metric. We randomly select $M = 10.000$ pairs of de-identified faces and check their similarity, to maintain tractability and avoid the evaluation of all possible pairs. Formally:

$$DIV(\tilde{\mathbf{X}}) = \frac{1}{M} \sum_{m=1}^{M} 1 - f_{rec}(\tilde{\mathbf{x}}_m^{(1)}, \tilde{\mathbf{x}}_m^{(2)}), \tag{14}$$

where $\tilde{\mathbf{x}}_m^{(\cdot)}$ denotes a member of the m-th pair. Thus, this metric measures the amount of de-identified pairs of faces which are recognized as different.

Although faceness and de-identification suffice in terms of quantitatively evaluating the de-identification results, we include diversity as a metric which quantizes the quality of the resulting faces in terms of realism of the photos after all faces have been de-identified.

4.2. Datasets

All of the datasets used to validate the proposed method have been annotated exhaustively either manually, or automatically, by algorithms that have been trained on manually annotated datasets with publicly disclosed information, such as ethnicity, age, gender, etc.

Labeled Faces in the Wild. For our experiments, we use the aligned LFW dataset [40], which consists of about 13.000 images containing faces as well as attributes for each image, including the depicted person's identity, gender, ethnicity, facial expression etc. About 1.600 of the different people are only depicted in a single image. As the images from the raw dataset contain faces in context, we use the face detector provided by dlib to extract tight bounding boxes and crop the detected faces to 64×64 grayscale images. Recent works on face recognition achieve results comparable to or even better than human-level performance [8], making the dataset a great candidate for face de-identification purposes. The face recognition system from dlib, which we use for the purpose of measuring the de-identification effect, achieves an accuracy of 99.38% on this dataset.

CelebA. We also use the CelebA dataset [43], containing over 200K images of about 10K different people. Like LFW, we use the dlib provided face detector to extract tight bounding boxes around the depicted faces and crop grayscale images of size 64×64. The baseline faceness and diversity for CelebA is 98.63% and 99.1% respectively.

4.3. Experimental Results

We experiment with both fully connected and fully convolutional models to compare both architectures. As the autoencoders tend to produce a blurred reconstruction, we also use a *deblurrer* network, trained to remove gaussian blur from images. This network is another, shallow autoencoder trained with blurred images and their uncontaminated counterparts as the target. The deblurrer is trained using images blurred with gaussian noise and the original images as their targets. Furthermore, as the deblurrer may introduce artifacts into the deblurred face, we make use of another autoencoder which acts as a *smoothener*. This network is a standard autoencoder, trained on facial images. As it has not seen such artifacts during training, the smoothener will inherently remove them from the reconstruction. It should be noted that the deblurrer and smoothener are quite shallow networks and the AE is the deepest network. Still, the entire architecture is much more compact than other state-of-the-art networks, making it very fast. Furthermore, the networks are trained separately as the goals of these two nets are more generic and not directly related to de-identification.

Figure 3 illustrates the proposed system architecture, with a fully convolutional autoencoder as the base de-identification network. The de-identifier can be a fully connected or fully convolutional autoencoder trained with the objective proposed in Section 3. In deployment, a single forward pass of the image suffices to produce a de-identified face. The deblurrer is implemented as a fully convolutional autoencoder with four layers and no downsampling factors. The output of the deblurrer is a sharpened de-identified face, although the process might introduce artifacts. Finally, the smoothener removes these artifacts and outputs the final de-identified face. We found that the use of the deblurrer in combination with the smoothener lead to increased de-identification rates. The deblurrer significantly affects the faceness of the resulting images and has a small effect on the de-identification rate. The smoothener has a strong effect on the diversity measure, which can be attribute to the removal of artifacts which may negatively effect the recognizer. The entire procedure can be viewed as a sequential array of neural networks.

Table 1: Ablation study into the network architecture's effect on various de-identification metrics for the fully connected model.

Encoder Architecture	*FCNS*	*DEID*	*DIV*
$4096 - 2048$	<u>74.53</u>	85.44	69.80
$4096 - 1024$	**77.23**	84.30	72.53
$4096 - 512$	*76.55*	84.74	70.33
$4096 - 1024 - 768$	73.43	*90.37*	*75.94*
$4096 - 1024 - 768 - 512$	74.44	<u>94.40</u>	<u>75.94</u>
$4096 - 2048 - 1024 - 762 - 512$	56.62	**97.38**	**78.42**

The final result is a more realistic face, which firstly boosts the faceness scores and secondly helps the face recognition system produce correct results by providing more face-like images.

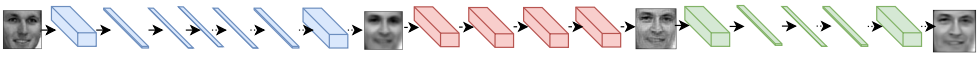

Figure 3: Deployment of the proposed de-identification system. The proposed modified autoencoder first de-identifies an input face. The output becomes the input to the deblurrer network, which produces a sharper face. Artifacts are removed by the smoothener. The entire system can be viewed as an end-to-end neural network sequence.

4.3.1. Fully Connected Model

We experiment first with a autoencoder which only uses fully-connected layers, i.e., the input image is flattened into a $64 \times 64 = 4096$-dimensional feature vector before it is forwarded to the network.

Training configuration. We perform an ablation study into the network architecture of a standard autoencoder, to find which one works the best for de-identification purposes while still producing images with high faceness scores. For this purpose, we fix the number of training epochs, and experiment with the depth and width of the model. The results of this study are summarized in Table 1, where the first column summarizes the architecture of the encoder — the decoder architecture is symmetrical. The results are obtained by training a standard autoencoder using the faces extracted from the LFW dataset and evaluating the model on a subset of the CelebA dataset consisting of 10K faces.

Table 2: Ablation study into the number of training epochs' effect on various de-identification metrics for the fully connected model.

Epochs	*FCNS*	*DEID*	*DIV*
150	74.74	94.00	66.77
200	**74.97**	**94.64**	74.51
300	74.46	*93.00*	*74.20*
400	*74.55*	92.64	**75.33**

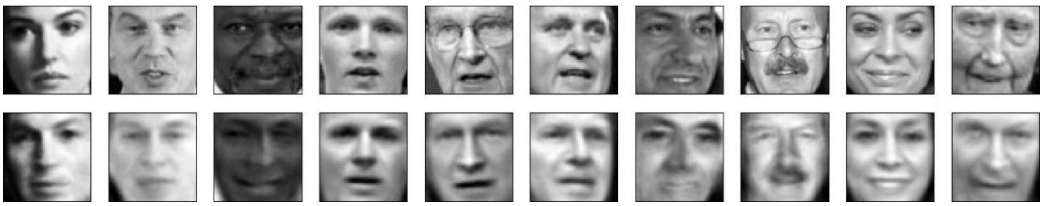

Figure 4: Examples of de-identified faces, as produced by a standard autoencoder trained on LFW, which achieves 94.64% de-identification and 74.55% faceness at the CelebA 10K test subset.

Although the best de-identification and diversity results are obtained using the deepest architecture, this can be purely attributed to the significantly lower faceness score. Thus, we choose the second deepest architecture $(4096 - 1024 - 768 - 512)$ as the one providing the best baseline de-identification score. We then examine the number of training epochs required to train this architecture, to achieve a good balance between the three metrics. The results are summarized in Table 2. We get the best result in terms of both faceness and de-identification at 200 training epochs. Although the diversity score increases with the number of epochs, the loss in faceness and de-identification justifies our choice of stopping the training process at 200 epochs.

Although the de-identification scores achieved by this standard autoencoder may seem encouraging, the resulting faces are of very low quality as corroborated by the low faceness scores and illustrated in Figure 4. Thus, a standard autoencoder cannot provide visually pleasing faces with identities different from the original ones.

Evaluation. Having settled on the network architecture and number of training epochs, we choose another subset of 10K faces from CelebA for the evaluation of the model.

Supervised De-identification We experiment with various settings of the attraction and repulsion matrices $\mathbf{P}$ and $\mathbf{Q}$, as defined by the attraction and repulsion sets of samples. We exploit the *ethnicity* (Ethn), *mood* (Mood) and *mouth opening* (Mth) as characteristics which accompany each face from the LFW dataset to train the encoder. For the supervised paradigm, we select the k-Nearest Neighbors of a sample with the same characteristics as the set of samples with desirable attributes $\mathcal{D}_i^n$, and the sample's k-Nearest Neighbors with different characteristics as the set of samples with conflicting attributes $\mathcal{C}_i^n$.

Table 3 presents a comparison between two experiments, one for $k = 10$ and one for $k = 20$ nearest neighbors used for the shift. We experiment both with a single shift involving the ethnicity of the depicted people, which is arguably their most prominent attribute, as well as with multiple gradual shifts towards different characteristics. While all methods improve upon the baseline faceness and de-identification, the diversity of the resulting faces decreases as the number of shifts increases. The decrease in diversity is due to the averaging process which takes place when computing the attraction and repulsion matrices, the rows of which become the mean of k samples for each shift. Despite the decrease in diversity, the performance gain in terms of faceness and de-identification is significant, and more prominent in the case of $k = 20$ neighbors. The diversity is also larger in general for this case, which can be attributed to the larger number of identities contributing to each de-identified face.

Table 3: Comparative results for $k = 10$ and $k = 20$ neighbors, in the supervised paradigm. Shifting towards multiple attributes increases the faceness and de-identification results but decreases the diversity of the de-identified faces.

	Shift Method	*FCNS*	*DEID*	*DIV*
	No shift	92.88	85.22	**59.47**
$k = 10$	Ethn	96.82	88.12	<u>59.27</u>
	Ethn+Mood	*99.10*	*91.88*	48.51
	Ethn+Mood+Mth	98.25	91.87	43.40
$k = 20$	Ethn+Mood	<u>99.76</u>	<u>92.36</u>	*51.44*
	Ethn+Mood+Mth	**99.85**	**92.65**	48.25

For successive shifts, the final attribute has the most prominent effect on the original sample. This is because the effect of the shift is gradually faded out — multiplied by $(1 - \alpha)$ — after each successive shift. Thus, we also

Table 4: Comparative results for various orders of shifting towards multiple attributes.

Shift Method	$FCNS$	$DEID$	DIV
Ethn+Mood+Mth	**99.85**	92.65	*48.25*
Ethn+Mth+Mood	99.78	92.56	48.96
Mood+Ethn+Mth	99.71	92.74	39.60
Mood+Mth+Ethn	*99.76*	*92.94*	**51.28**
Mth+Mood+Ethn	99.52	93.02	41.08
Mth+Ethn+Mood	99.62	**93.22**	34.00

experiment with the order of the shifts, for the $k = 20$ case. The results of this experiment are summarized in Table 4. The mouth opening and mood attributes mostly contribute to the general faceness of a face, i.e., how well it resembles a natural face and thus is recognizable as one by a face detector. Thus, if the samples are shifted towards these attributes last, the network should produce de-identified faces which resemble natural faces the most. This is corroborated by the faceness scores, which are the highest for these cases.

Furthermore, as aforementioned, the ethnicity of the depicted people is their most prominent attribute. Hence, intuitively, it should contribute the most to the shift in order to achieve de-identification and diverse faces. As shown in Table 4, the highest de-identification scores are achieved when ethnicity is the last or next-to-last attribute to contribute to the shift.

Unsupervised De-identification For the unsupervised paradigm, we use the k-Means algorithm to first uncover clusters in the low-dimensional subspace learned by the autoencoder and define sets $\mathcal{D}_i^{\mathcal{U}}$ and $\mathcal{C}_i^{\mathcal{U}}$. Then, each sample is shifted towards the center of the cluster it belongs to. For clusters with a large number of samples, all of the clusters samples will be shifted towards the same target. This will naturally lead to low diversity scores. This is illustrated in Figure 5 and Table 5, which presents the faceness, de-identification and diversity scores for various numbers of clusters K.

On the contrary, for smaller clusters, there should be more diversity in the de-identified faces due to the larger availability of cluster centers to shift towards. Whereas in Figure 5, the attributes of the faces have been wiped out by the large number of samples contributing to their shift, in Figure 6, where each cluster consists of much fewer samples, most of the attributes

Table 5: Comparative results for shift towards the center of the closest cluster for various number of clusters uncovered by k-Means.

K	FCNS	DEID	DIV
10	**100**	**97.74**	7.42
50	**100**	95.76	*27.37*
180	99.83	*95.10*	**44.93**
300	*99.53*	94.91	44.77

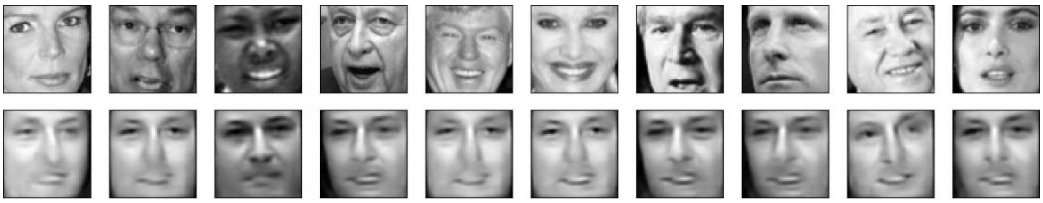

Figure 5: Examples of de-identified faces, for $K = 10$ clusters. All samples have shifted towards the same target, all of which have very similar features.

of the original faces are preserved after the de-identification process. This includes the ethnicity and mouth opening attributes, whereas in Figure 5, only the pose is preserved, which may be attributed to the k-Means algorithm clustering faces into group with more or less the same pose.

In comparison to the supervised methods, unsupervised de-identification leads to lower diversity scores, undoubtedly due to the fact that multiple samples will have the same target (i.e., cluster center). As the number of clusters increases, so does the resulting diversity. However, the highest diversity achieved using the unsupervised method is still much lower than the highest diversity achieved using supervised methods (44.93 and 51.28 respectively). However, the best setting in the unsupervised scenario yields slightly better faceness and significantly better de-identification, somewhat sacrific-

Figure 6: Examples of de-identified faces, for $K = 180$ clusters. The de-identified faces are visibly and quantitatively more diverse.

ing naturalness for the sake of more effective de-identification.

4.3.2. Fully Convolutional Model

Training configuration. As in the fully connected model, we first perform a study into the most suitable network architecture. We make use of max pooling layers with stride 2 to quickly downsample the 64×64 input grayscale image into a volume with as few channels as possible, while maintaining low reconstruction errors. Finally, we chose an architecture which alternates between convolutional layers with 3×3 filters and max pooling layers with stride 2, until the input is downsampled into a 2×2 volume with 256 channels. In this representation, each grid cell roughly corresponds to a spatially corresponding facial attribute, i.e., left eye, right eye, left cheek, right cheek. Thus we can choose to either shift the entire representation by first flattening it into a $2 \times 2 \times 256 = 1024$-dimensional vector, or to shift each feature vector extracted for each grid cell towards feature vectors extracted at the same spatial location.

In our experiments with convolutional autoencoders, we found that they are more likely to create artifacts caused by misalignment of the faces seen during its training stage and faces used for evaluation. In general, the faces in CelebA are not as well aligned as the ones in LFW. For this reason, we incorporate 10K faces from CelebA into the training set of the autoencoder, and focus entirely on unsupervised de-identification methods.

Evaluation. We evaluate various settings in the same set of 10K faces from CelebA as the previous experiments. We experiment first with flattening the learned representations and shifting them towards one neighbor to ensure crisp de-identification. We choose this neighbor out of a set of n neighbors to be the one which lies the closest to the representation of the mean face, thus only defining the set $\mathcal{D}_i^{\mathcal{U}}$ of samples with desirable properties. We simultaneously perform a study into the effect of the number of neighbors n as well as the hyperparameter α controlling the degree of the shift towards this neighbor.

The results are summarized in Table 6 for two different values of n and two values of α. The best faceness and de-identification scores are achieved for $n = 9$ neighbors, although all settings lead to almost perfect faceness. This isn't true for the de-identification results, which are significantly improved by increasing the number of neighbors. However, as the number of neighbors increases, the diversity of the faces decreases. Although the shift is performed

using only one of those neighbors, the larger the number of neighbors in which we search for this sample, the higher the chance that multiple samples will be assigned the same target. This is evident by the decrease in diversity score both as n increases and as α, the degree of the shift, increases. The smaller the value of α is, the closer each sample remains to its original representation. The higher the value, de-identification increases but diversity decreases as shifted samples are clustered together. Effectively, α controls the similarity between the original and de-identified faces in the latent space uncovered by the autoencoder.

Table 6: Study into the effect of the hyperparameters on the de-identification results.

		FCNS	*DEID*	*DIV*
$\alpha = 0.4$	$n = 5$	98.66	56.14	**87.87**
	$n = 7$	98.92	72.46	78.00
	$n = 9$	*99.51*	<u>91.48</u>	66.08
$\alpha = 0.8$	$n = 5$	98.87	59.25	<u>86.75</u>
	$n = 7$	99.15	76.43	76.72
	$n = 9$	**99.80**	**93.97**	55.14

For the case where each grid cell feature vector, roughly corresponding to a facial characteristic, is shifted independently of each other, we choose to shift each one towards its n-th nearest neighbor, without taking into consideration the mean representation for that grid cell. This is to account for the higher diversity present in distinct facial characteristics when the faces to be de-identified are misaligned. For example, if a person is posed slightly to the left, it is counter intuitive to shift their upper left characteristic towards those of people who are depicted in a frontal view. The sample's n-th neighbor, however, is very likely to be similar in pose and structure, but dissimilar in identity. Figure 7 shows examples of this variant on color images, the first row showing the original faces, whereas the reconstructions from a standard AE and a proposed AE are shown in the second and third rows respectively.

The results are presented in Table 7, for various values of n. We start by moving each feature vector towards its closest neighbor, which yields the highest diversity but very low de-identification. The best results are given for $n = 20$ neighbors, in terms of de-identification, with also high faceness and diversity scores. This is in contrast with the results presented in Table 6. This is because, for each grid cell feature vector, its target representation

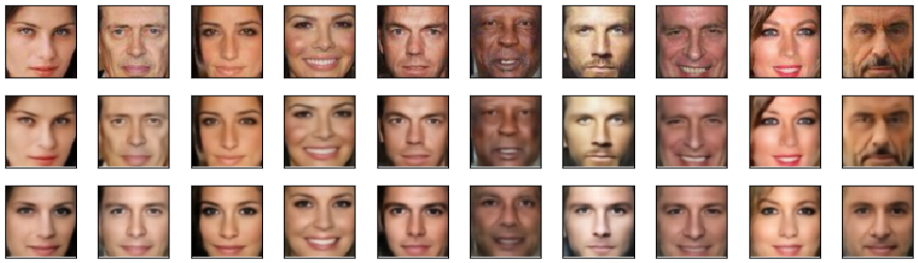

Figure 7: Examples of color images. First row shows the original faces, second one the reconstructed ones by a standard AE, and third row shows de-identified faces using the proposed method.

Table 7: Comparative results for the unsupervised paradigm, where each grid cell feature is shifted towards its n-th closest neighbor.

	FCNS	*DEID*	*DIV*
$n = 1$	98.04	*59.72*	**88.76**
$n = 5$	<u>98.48</u>	<u>61.29</u>	*84.93*
$n = 10$	**98.62**	57.55	<u>87.43</u>
$n = 20$	*98.15*	**65.11**	84.45

may belong to a different identity. In total, up to four identities may be combined to produce the final de-identified face.

In comparison to the fully connected models, the convolutional networks yield significantly higher diversity scores at the cost of slightly reduced de-identification scores — more significantly in the case where each spatial feature is modified separately. Figure 8 illustrates examples of new faces generated by shifting each spatial feature towards its 10th closest neighbor. The generated faces are quite visually pleasing and maintain many of the original face's properties, such as expression, gender, pose and facial accessories. However, this realism and diversity leads to decreased de-identification rates.

From the results presented in Tables 3-7, it is clear that the fully connected models offer better de-identification rates, even in the unsupervised scenario, while sacrificing the diversity of the reconstructed faces in comparison to the fully convolutional models. The faceness scores are similar in both cases. Furthermore, the unsupervised experiments indicate that AEs are capable of extracting representations which implicitly incorporate attribute information, allowing the proposed methods to work effectively even in the absence of attribute labels. Deciding on a variant depends on the desired

Figure 8: Examples of de-identified faces, for modification towards the 10-th neighbor.

application-specific de-identification to diversity trade-off.

Although the used datasets mostly contain frontal images, the proposed methods are designed to work with any facial pose, as this can be considered a preservable attribute and added to the list of attracting attributes to force the samples to maintain the original pose. This is already implicitly imposed by the autoencoder, which even in the unsupervised training, learns to map images with similar attributes close together. Given appropriate training samples, the proposed method can handle all variations of the attributes that appear in the training dataset.

4.4. Anonymization & Attribute Preservation

We finally perform an experiment on the subset of 200 people used in [12], to compare the two methods and measure the anonymization and attribute preservation capability of the proposed method. Our results, using a fully connected AE model, with the encoded representations being shifted towards their cluster center, are summarized in Table 8. We choose this variant of the proposed method for its high faceness scores and its relatively low diversity scores, which will lead to highly usable anonymized data. We measure the anonymization (ANON), following k-Anonymity theory [10], i.e., by counting the minimum number of faces from the gallery (test) set that are indistinguishable from one another.

Furthermore, we evaluate the attribute preservation ability of the proposed method, by calculating the micro-averaged Precision score over four attributes: one for gender and three for ethnicity (ATTR). We consider the attributes produced by an attribute classifier to be the groundtruth attributes[2].

[2]https://github.com/wondonghyeon/face-classification

Table 8: Deidentification, anonymization and attribute preservation for a subset of 200 people used in [12], using a fully connected AE. The best variant of the competitive method achieves a de-identification rate of about 90%.

K	DEID	ANON	ATTR
10	99%	39	58%
50	99%	81	55%
180	100%	55	49%
300	99%	49	55%

The proposed method attains perfect de-identification scores while achieving very high anonymization rates. Furthermore, based on the micro-averaged precision scores, the proposed method seems to sufficiently preserve the attributes of the original faces.

4.5. Speed & Deployment on Embedded Devices

Due to the lightweight architecture of the proposed de-identification pipeline, including the deblurring and smoothening networks, it is capable of running on embedded systems. We investigate the speed of the proposed method on an NVIDIA Jetson TX2 module, to facilitate de-identification on videos captured by unmanned robots such as UAVs The Jetson TX2 is a very lightweight computer with a CUDA-enabled GPU, which supports fast computation of calculations, such as those performed by convolutional neural networks. De-identification on such videos is crucial, as flights on public areas raise several privacy concerns. For an input face of size 64×64, a de-identified face is produced at 0.65ms, while achieving very high de-identification rates and photorealistic results, as illustrated in the previous sections. Thus, in combination with a fast face detector [44, 30], a privacy preserving system based on face de-identification can run at real-time speed even on systems with limited computational resources.

5. Conclusions

We have presented multiple methods for face de-identification based on fully connected and fully convolutional autoencoders, by training the encoder to learn to shift the faces towards other faces with desirable attributes and away from samples with conflicting attributes. More specifically, we have presented various ways to acquire new encoding targets in both a supervised and unsupervised setting. When attribute labels are available, we have

presented straightforward ways to incorporate this information into the encoder so as to produce faces which are de-identified while maintaining their faceness, i.e., their ability to be recognized as faces. Even when labels are unavailable, which is true for the majority of data massively available, we have proposed various intuitive ways to train the encoder to achieve high de-identification and faceness scores. Moreover, we introduce the diversity metric, to quantize the quality of the produced faces in terms of the aesthetic result, so that a photo with de-identified faces will remain as natural as possible while achieving its purpose of maintain the anonymity of those depicted in it. Finally, due to the simplicity of the proposed de-identification system, it can be deployed on embedded devices. Thus it can be efficiently used to address privacy concerns in scenarios where privacy preservation is crucial, such as in autonomous UAV flights.

6. Acknowledgments

This project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 731667 (MULTIDRONE). This publication reflects the authors views only. The European Commission is not responsible for any use that may be made of the information it contains.

References

[1] T. Z. Zarsky, Incompatible: The gdpr in the age of big data, Seton Hall L. Rev. 47 (2016) 995.

[2] V. Bruce, A. Young, Understanding face recognition, British journal of psychology 77 (3) (1986) 305–327.

[3] S. Joon Oh, R. Benenson, M. Fritz, B. Schiele, Person recognition in personal photo collections, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3862–3870.

[4] E. M. Newton, L. Sweeney, B. Malin, Preserving privacy by de-identifying face images, IEEE transactions on Knowledge and Data Engineering 17 (2) (2005) 232–243.

[5] S. J. Oh, R. Benenson, M. Fritz, B. Schiele, Faceless person recognition: Privacy implications in social media, in: European Conference on Computer Vision, Springer, 2016, pp. 19–35.

[6] P. Sinha, B. Balas, Y. Ostrovsky, R. Russell, Face recognition by humans: Nineteen results all computer vision researchers should know about, Proceedings of the IEEE 94 (11) (2006) 1948–1962.

[7] S. Yang, P. Luo, C. C. Loy, X. Tang, Faceness-net: Face detection through deep facial part responses, IEEE transactions on pattern analysis and machine intelligence 40 (8) (2018) 1845–1859.

[8] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 815–823.

[9] S. Ribaric, A. Ariyaeeinia, N. Pavesic, De-identification for privacy protection in multimedia content: A survey, Signal Processing: Image Communication 47 (2016) 131–151.

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

[11] R. Gross, L. Sweeney, F. De la Torre, S. Baker, Model-based face de-identification, in: null, IEEE, 2006, p. 161.

[12] A. Jourabloo, X. Yin, X. Liu, Attribute preserved face de-identification., in: ICB, Citeseer, 2015, pp. 278–285.

[13] T. Li, L. Lin, Anonymousnet: Natural face de-identification with measurable privacy, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2019, pp. 0–0.

[14] L. Tran, X. Yin, X. Liu, Disentangled representation learning gan for pose-invariant face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1415–1424.

[15] J. Zhao, Y. Cheng, Y. Xu, L. Xiong, J. Li, F. Zhao, K. Jayashree, S. Pranata, S. Shen, J. Xing, et al., Towards pose invariant face recognition in the wild, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2207–2216.

[16] W. Xu, K. Shawn, G. Wang, Toward learning a unified many-to-many mapping for diverse image translation, Pattern Recognition 93 (2019) 570–580.

[17] K. Brkić, T. Hrkać, I. Sikirić, Z. Kalafatić, Towards neural art-based face de-identification in video data, in: Sensing, Processing and Learning for Intelligent Machines (SPLINE), 2016 First International Workshop on, IEEE, 2016, pp. 1–5.

[18] Q. Sun, L. Ma, S. J. Oh, L. Van Gool, B. Schiele, M. Fritz, Natural and effective obfuscation by head inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5050–5059.

[19] B. Meden, R. C. Mallı, S. Fabijan, H. K. Ekenel, V. Štruc, P. Peer, Face deidentification with generative deep neural networks, IET Signal Processing 11 (9) (2017) 1046–1054.

[20] K. Brkic, I. Sikiric, T. Hrkac, Z. Kalafatic, I know that person: Generative full body and face de-identification of people in images, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. IEEE, 2017, pp. 1319–1328.

[21] Z. Ren, Y. Jae Lee, M. S. Ryoo, Learning to anonymize faces for privacy preserving action detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 620–636.

[22] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the International Conference on Machine Learning, 2008, pp. 1096–1103.

[23] S. Gao, Y. Zhang, K. Jia, J. Lu, Y. Zhang, Single sample face recognition via learning deep supervised autoencoders, IEEE Transactions on Information Forensics and Security 10 (10) (2015) 2108–2118.

[24] P. Nousi, A. Tefas, Deep learning algorithms for discriminant autoencoding, Neurocomputing 266 (2017) 325–335.

[25] M. Kan, S. Shan, H. Chang, X. Chen, Stacked progressive auto-encoders (spae) for face recognition across poses, in: Proceedings of the IEEE

Conference on Computer Vision and Pattern Recognition, 2014, pp. 1883–1890.

[26] J. Zhang, M. Kan, S. Shan, X. Chen, Occlusion-free face alignment: deep regression networks coupled with de-corrupt autoencoders, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3428–3437.

[27] X. Yu, F. Porikli, Hallucinating very low-resolution unaligned and noisy face images by transformative discriminative autoencoders, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3760–3768.

[28] W. Xu, S. Keshmiri, G. R. Wang, Adversarially approximated autoencoder for image generation and manipulation, IEEE Transactions on Multimedia.

[29] W. Xu, S. Keshmiri, G. Wang, Stacked wasserstein autoencoder, Neurocomputing 363 (2019) 195–204.

[30] D. Triantafyllidou, P. Nousi, A. Tefas, Fast deep convolutional face detection in the wild exploiting hard sample mining, Big data research 11 (2018) 65–76.

[31] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[32] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng, Multimodal deep learning, in: Proceedings of the International Conference on Machine Learning, 2011, pp. 689–696.

[33] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Cognitive modeling 5 (3) (1988) 1.

[34] T. Zhang, Solving large scale linear prediction problems using stochastic gradient descent algorithms, in: Proceedings of the International Conference on Machine Learning, 2004, p. 116.

[35] J. Snoek, R. P. Adams, H. Larochelle, Nonparametric guidance of autoencoder representations using label information, Journal of Machine Learning Research 13 (Sep) (2012) 2567–2588.

[36] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199.

[37] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: International conference on machine learning, 2016, pp. 478–487.

[38] P. Nousi, A. Tefas, Self-supervised autoencoders for clustering and classification, Evolving Systems (2018) 1–14.

[39] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft, When is nearest neighbor meaningful?, in: International conference on database theory, Springer, 1999, pp. 217–235.

[40] E. Learned-Miller, G. B. Huang, A. RoyChowdhury, H. Li, G. Hua, Labeled faces in the wild: A survey, in: Advances in face detection and facial image analysis, Springer, 2016, pp. 189–248.

[41] I. Jolliffe, Principal component analysis, Springer, 2011.

[42] D. E. King, Dlib-ml: A machine learning toolkit, Journal of Machine Learning Research 10 (2009) 1755–1758.

[43] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV), 2015.

[44] P. Nousi, E. Patsiouras, A. Tefas, I. Pitas, Convolutional neural networks for visual information analysis with limited computing resources, in: 2018 25th IEEE International Conference on Image Processing (ICIP), IEEE, 2018, pp. 321–325.