Machine Learning with Applications 10 (2022) 100412

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Evaluating a face generator from a human perspective

Joris Pries [a,*], Sandjai Bhulai [b], Rob van der Mei [a]

[a] Centrum Wiskunde & Informatica, Department of Stochastics, Science Park 123, Amsterdam 1098 XG, Netherlands
[b] Vrije Universiteit, Department of Mathematics, De Boelelaan 1111, Amsterdam 1081 HV, Netherlands

ARTICLE INFO

Keywords:
Face generator
StyleGAN2
Attribute prediction
Facial recognition
Clustering
Truncation

ABSTRACT

StyleGAN2 is able to generate very realistic and high-quality faces of humans using a training set (*FFHQ*). Instead of using one of the many commonly used metrics to evaluate the performance of a face generator (e.g., *FID*, *IS* and *P&R*), this paper uses a more humanlike approach providing a different outlook on the performance of StyleGAN2. The generator within StyleGAN2 tries to learn the distribution of the input dataset. However, this does not necessarily mean that higher-level human concepts are preserved. We examine if general human attributes, such as age and gender, are transferred to the output dataset and if StyleGAN2 is able to generate actual new persons according to facial recognition methods. It is crucial for practical implementations that a face generator not only generates new humans, but that these humans are not clones of the original identities. This article addresses these questions. Although our approach can be used for other face generators, we only focused on StyleGAN2. First, multiple models are used to predict general human attributes. This shows that the generated images have the same attribute distributions as the input dataset. However, if truncation is applied to limit the latent variable space, the attribute distributions change towards the attributes corresponding with the latent variable used in truncation. Second, by clustering using face recognition models, we demonstrate that the generated images do not belong to an existing person from the input dataset. Thus, StyleGAN2 is able to generate new persons with similar human characteristics as the input dataset.

1. Introduction

Think of an unknown face. Humans are capable of imagining faces they have never seen before, combining facial attributes from multiple sources to create a new identity. Can a machine do the same? By looking at real images of humans, can it learn how to generate a unique and realistic face? And if so, are humans still able to distinguish between authentic and computer-generated faces? These questions are part of a larger quest of discovering the capabilities and boundaries of machines. Speech, music, paintings, images, and even videos are among the many things a computer is now able to generate. The quality of the produced content has increased rapidly since the introduction of generative adversarial networks (GAN Goodfellow et al., 2014). Generating realistic faces shows the power, capabilities, and limitations of these approaches.

In 2019, the successor (Karras et al., 2019) of the well-known StyleGAN (Karras, Laine, & Aila, 2018) paper was published. When StyleGAN (Karras et al., 2018) was released in 2018, it immediately showed impressive results. At that time, this architecture improved the state-of-the-art performance considerably by injecting the generator at different stages with a style-based latent variable. Although humans can still distinguish between computer-generated and real images, the images look very realistic at first glance. This is a huge achievement. Especially if one considers that only since 2017, Karras, Aila, Laine, and Lehtinen (2017) were able to generate high-resolution images (1024 × 1024 pixels). Only small details give away that these images are not real (West & Bergstrom, 2019). The successive paper (Karras et al., 2019) claims to improve the images even further, making them even less distinguishable. Their new approach is called StyleGAN2.

As the name suggests, StyleGAN2 is trained using a generative adversarial network (GAN) (Goodfellow et al., 2014). The basis of this approach is to let two models compete against each other, making each model better at their specific task. More specifically, one model tries to generate images that resemble real faces, whereas the other model tries to distinguish between the real and the generated images. The generator within StyleGAN2 tries to learn the distribution of the input images, which is monitored using the metrics *FID*, *P&R*, and *PPL*. According to these metrics, StyleGAN2 is successful in learning the input distribution. However, this does not necessarily mean that higher-level human concepts are preserved. Are the input and generated images similar from a human perspective?

When a human compares two faces, common measures for evaluating GANs like *FID* (Heusel et al., 2017), *IS* (Salimans et al., 2016) or *PPL* (Karras et al., 2018) are not natural, as these measures are artificially using e.g., a neural network to evaluate the performance.

[*] Corresponding author.
E-mail addresses: joris.pries@cwi.nl (J. Pries), s.bhulai@vu.nl (S. Bhulai), mei@cwi.nl (R. van der Mei).

https://doi.org/10.1016/j.mlwa.2022.100412
Received 8 February 2022; Received in revised form 16 August 2022; Accepted 5 September 2022
Available online 9 September 2022
2666-8270/© 2022 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

This is not a human approach, a person would rather compare human characteristics to evaluate the images. Although it is infeasible to compare a lot of generated images by hand, a humanlike approach is necessary. Zhou et al. (2019) did use humans to decide whether images generated by StyleGAN were fake or real. The results showed that StyleGAN was capable of generating faces that were hard to distinguish by humans from the input images. Our research focuses on two different aspects: Are human traits transferred from the input to the output dataset and are the generated images new identities? Lack of attribute and identity labels tagged by humans for StyleGAN2, leads us to use existing models that were trained using different humanly labeled data.

Hence, we take two separate paths to evaluate how well StyleGAN2 performs from a 'human' perspective. First, multiple models are used to predict general attributes of the images, such as age, gender, and race. In this way, we can determine if higher-level concepts are preserved. Second, we examine for multiple *face recognition* models if the generated images can be considered to be different persons, compared to the images from the input dataset.

With this two-pronged approach, we are able to show that the human attribute distributions are very similar for the input and output dataset, but the generated images are nonetheless different according to the facial recognition models. Thus, StyleGAN2 has the best of two worlds. It is able to copy high-level concepts from the input dataset, whilst still creating different persons. Furthermore, if *truncation* (see Section 2.1) is used to limit the latent variable space, the attribute distributions change significantly towards the attributes corresponding with the latent variable used in truncation. To our knowledge, this humanly approach to comparing high-level concepts of facial datasets is new. While we will only use our two-pronged approach for StyleGAN2, it can also be used to evaluate other face generators.

To summarize, in this paper we:

- introduce a new two-pronged humanly approach to evaluate face generators, by predicting human attributes and clustering using face recognition models;
- show that the state-of-the-art StyleGAN2 generates images that have the same attribute distributions as the input dataset;
- determine that StyleGAN2 generates faces that often do not belong to persons in the input dataset according to face recognition models;
- observe that adding truncation to the latent variable space changes the attribute distributions towards the attributes corresponding with the latent variable used in truncation.

The remainder of this paper is organized as follows. First, the relevant datasets are discussed in Section 2. Next, in Section 3 the methods are explained that are used to predict facial attributes, embed faces, and cluster on these embeddings. Furthermore, we also define how a cluster is evaluated and why clustering is a natural approach. The results are discussed in Section 4. Finally, Section 5 summarizes the general findings and discusses possible future research opportunities.

2. Datasets

Karras et al. (2019) made three datasets publicly available that are used in this research: The input dataset (*FFHQ*), consisting of 70,000 real facial images (without identity annotation); Two output datasets of StyleGAN2, both consisting of 100,000 generated images. The only difference in the creation of these output datasets is the so-called *truncation* (Brock, Donahue, & Simonyan, 2018; Karras et al., 2018, 2019) parameter. All images are high-quality pictures (1024 × 1024 pixels).

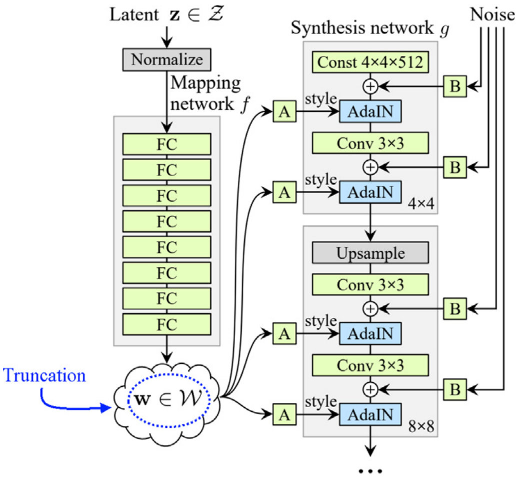

Fig. 1. StyleGAN structure: Architecture of the StyleGAN approach (extracted from Karras et al. (2018)). Truncation limits the intermediate latent space $\mathcal{W}$.

2.1. Truncation

To explain how truncation works, it is useful to take a look at the structure of StyleGAN (see Fig. 1). Note that there are some differences with the architecture of StyleGAN2. However, the following core principles still hold. Some latent variable $\boldsymbol{z} \in \mathcal{Z}$ from latent space $\mathcal{Z}$ goes into a mapping network f, after it is normalized using pixelwise feature vector normalization.[1] This results in a different variable $\boldsymbol{w} \in \mathcal{W}$ such that $f(\boldsymbol{z}) = \boldsymbol{w}$. $\mathcal{W}$ is the so-called *intermediate latent space*. Next, the expectation of the intermediate latent variable is determined by $\bar{\boldsymbol{w}} := \mathbb{E}_{\boldsymbol{z}\sim P(\boldsymbol{z})}[f(\boldsymbol{z})]$, where $P(\boldsymbol{z})$ is the probability that $\boldsymbol{z}$ is randomly drawn from $\mathcal{Z}$. The authors of Karras et al. (2018) state that $\bar{\boldsymbol{w}}$ represents "a sort of an average face".

$\bar{\boldsymbol{w}}$ is used to truncate the intermediate latent space. Given a $\boldsymbol{w} \in \mathcal{W}$, truncation returns a different intermediate latent variable, denoted $\boldsymbol{w}'$, such that $\boldsymbol{w}' = \bar{\boldsymbol{w}} + \psi \cdot (\boldsymbol{w} - \bar{\boldsymbol{w}})$, where $\psi \in \mathbb{R}$ is called the *truncation parameter*. Note that $\psi = 1$ gives $\boldsymbol{w}' = \boldsymbol{w}$, which is the same as not applying truncation at all. In Fig. 2, five faces are shown that are generated with $\bar{\boldsymbol{w}}$ as intermediate latent variable. This is equivalent to generating images with $\psi = 0$. Furthermore, noise is injected in the synthesis network to increase stochasticity, see Fig. 1. However, this leads to only minor changes if the intermediate latent variable is constant. As can be seen in Fig. 2, the faces all look very similar.

3. Methodology

To compare the input images of a face generator with its output, two separate paths are taken. First, multiple models are used to predict human attributes. This allows for a high-level comparison between the different datasets. Are characteristics, such as age and gender, the same for the input and output datasets? Secondly, clustering using face recognition models could determine if the generated faces belong to an existing person from the input dataset. Do the output datasets consist of different persons, or are they embedded similarly compared to the input dataset? Combining these two approaches gives a clear view of the performance of a face generator.

The output of StyleGAN has already been examined to some extent. Karras et al. (2019) evaluated the generated images in order to eliminate artifacts. For different datasets, *FID* and *PPL* was compared between StyleGAN and StyleGAN2 (Karras et al., 2018). Furthermore, efforts have been made to understand and steer the latent space (Shen,

[1] https://github.com/NVlabs/stylegan/blob/03563d18a0cf8d67d897cc61e44479267968716b/training/networks_stylegan.py.

Fig. 2. Average intermediate latent space: These faces (seeds 0000–0004) are generated using the expected intermediate latent variable $\bar{\boldsymbol{w}}$.

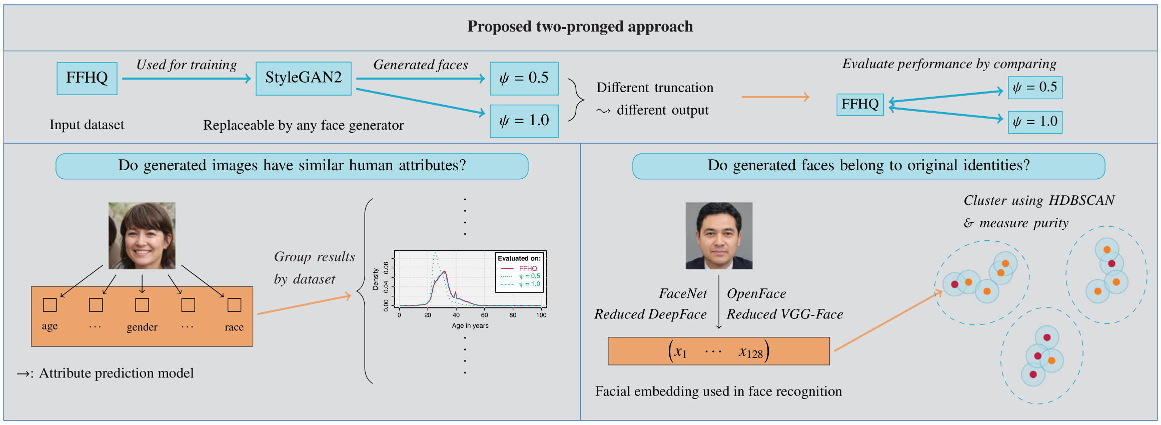

Fig. 3. General concept: A general overview of our proposed two-pronged approach. Using different truncation parameters, a comparison is made (with attribute prediction models and facial embedding methods) between the input and output datasets in order to determine if generated images have similar human attributes and if they belong to original identities.

Yang, Tang, & Zhou, 2020). By manipulating the latent space, one could change certain attributes of an image. For example, Shen et al. (2020) showed that it is possible to alter the age, gender, smile, pose, and add or remove eyeglasses. However, to our knowledge, our humanly approach to comparing high-level concepts of facial datasets is new. While we will only look at datasets from StyleGAN2, our two-pronged approach can also be used to evaluate other face generators. Within our novel approach, existing methods are used for predicting, embedding and clustering. These methods will all be discussed in the upcoming sections. A general overview of our proposed approach can be found in Fig. 3.

3.1. *Attribute prediction*

A face has many characteristics. This leads to a wide variety of attribute predictions: pose (Hu, Chen, Zhou, & Zhang, 2004), skin color (Vezhnevets, Sazonov, & Andreeva, 2003), and even attractiveness (Xu et al., 2017), to name just a few. We select the following group of features to examine: *age*, *gender*, *race*, *horizontal rotation*, and *vertical rotation*. Note that these features cover general human concepts, but additional attribution models can always be added to or removed from this framework. To clarify what we mean by 'human concept', we argue that every identifiable aspect of a face (or body) that has been given a name can be called a human concept. For example, the eyebrow is identified by humans as a specific part of the face. But also, somebody can look young or old. We consider these examples as 'human concepts', because we abstract information from a group of pixels with respect to some convention. In our view, any model that predicts a general human attribute, trained with humanly labeled data, could give some insight into the difference between the input and output dataset. Adding more attribute models does give additional information, but to show how our approach works, we limit ourselves to the attribute prediction models that we introduce in the following sections. Note that adding or removing other attribute prediction models does not affect the results of an individual attribute model, as each model is assessed separately.

3.1.1. *Predicting age, gender, and race*

One of the main guiding papers for this research is the *Diversity in Faces* paper by Merler, Ratha, Feris, and Smith (2019). The aim of their research was to create an annotated dataset in order to improve the accuracy of face recognition and increase the facial diversity within commonly used datasets. Lack of diversity could harm the effectiveness of face recognition in practical implementations. It could even be discriminatory against minorities (Buolamwini & Gebru, 2018). Merler et al. (2019) use different models to annotate images from the YFCC-100M dataset (Thomee et al., 2016). These models predict a plethora of attributes for each face. The same kind of models, implemented in *deepface* (Serengil & Ozpinar, 2020), are used to predict the *age*, *gender*, and *race* of a person.

However, there is a difference between the implementation of *deepface* (Serengil & Ozpinar, 2020) and the prediction models from Merler et al. (2019). *deepface* uses the VGG-Face neural network (Parkhi, Vedaldi, & Zisserman, 2015), whereas Merler et al. (2019) follows the approach of Rothe, Timofte, and Van Gool (2018), who use the VGG-16 architecture (Simonyan & Zisserman, 2014). The VGG-Face network (Parkhi et al., 2015) is specifically trained to recognize faces, whereas the VGG-16 network is trained with ImageNet (Russakovsky et al., 2015) by Rothe et al. (2018). ImageNet contains a wide variety of images, not limited to faces. This is why we decided to follow *deepface* and use the VGG-Face network.

For each attribute, a similar procedure is followed. *deepface* uses a pre-trained VGG-Face network (Parkhi et al., 2015) as the starting point. Only the last few layers are replaced and retrained to fit the objective. There are some important details about these models (see Serengil and Ozpinar (2020) for technicalities):

- Counterintuitively, age prediction is not made using regression. Rothe et al. (2018) claim that using classification instead of regression improved the performance and also stabilized the training process. The output layer consists of 101 variables, each corresponding to an age in years (0–100). The last layer has a softmax activation function, which ensures that the output of the last layer is a probability distribution over the different output variables. The age is finally predicted by taking the expectation over these output variables, see Rothe et al. (2018).

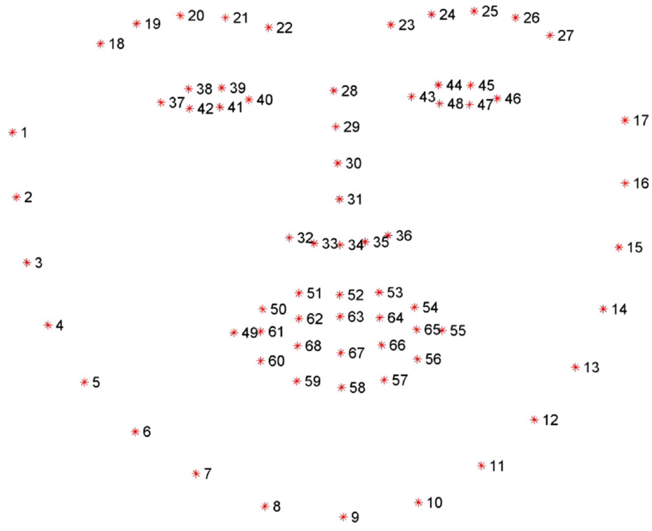

Fig. 4. Dlib landmarks: Dlib predicts the coordinates of these 68 landmarks for each face.
Source: extracted from Sagonas, Tzimiropoulos, Zafeiriou, and Pantic (2013).

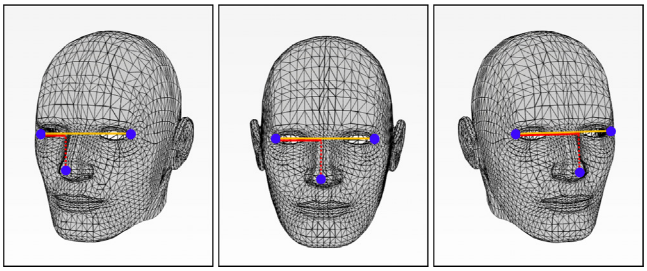

Fig. 5. Horizontal rotation: Horizontal rotation is measured by dividing the red bar by the yellow bar (see Eq. (1)). The positions of the blue dots are predicted by dlib. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

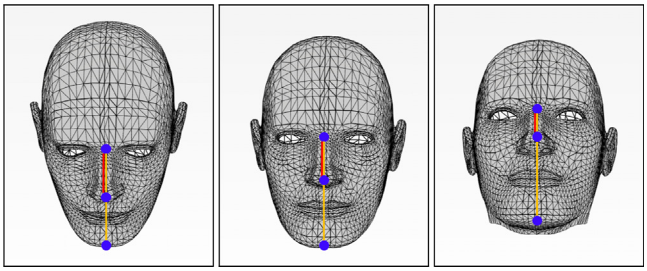

Fig. 6. Vertical rotation: Vertical rotation is measured by dividing the red bar by the yellow bar (see Eq. (2)). The positions of the blue dots are predicted by dlib. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- Gender prediction is made using two output variables, corresponding to *woman* and *man*.
- For race prediction, a distinction is made between the following races: *Asian*, *Indian*, *Black*, *White*, *Middle Eastern*, and *Latino Hispanic*.
- *deepface* uses the *haarcascade frontalface default* detector from OpenCV (Bradski, 2000) to center, trim, and resize an image. However, it can occur that the facial detector does not recognize a face. When this happens, the image is simply omitted from the analysis of the corresponding attributes.

Serengil and Ozpinar (2020) self-reported on the performance of the models. The mean absolute error of the age model was 4.65 and the accuracy of the gender model was 97.44% with 96.29% precision and 95.05% recall. However, the models were not evaluated on the datasets that will be used in this research, because there exist no annotated labels of these features yet. It is therefore unclear how well these models perform for the datasets that are used. Nevertheless, we want to stress the fact that these models are only used to compare the characteristics of each dataset globally. Even if the models perform worse (due to domain shift), they can still be insightful for comparing the datasets.

3.1.2. *Predicting horizontal and vertical rotation*

To measure the horizontal and vertical rotation, *dlib* (King, 2009) is used. It can predict the position of 68 general landmarks on a face (see Fig. 4). These landmarks can be used to crop an image or measure attributes such as face and nose width/height. We use the landmarks to estimate the horizontal and vertical position of a head. It must be noted that these points remain a prediction. Especially when a head is rotated too much, these predictions lose accuracy.

There are many ways to estimate the horizontal rotation (yaw) and vertical rotation (pitch) of a head (Breitenstein, Kuettel, Weise, van Gool, & Pfister, 2008; Díaz Barros, Mirbach, Garcia, Varanasi, & Stricker, 2019; Hu et al., 2004). However, we are mainly interested in the differences between the datasets. Therefore, we are not much concerned about obtaining the best accuracy for each individual image. Thus, we use a simple concept to estimate the horizontal and vertical rotation, see Figs. 5 and 6.

Let x_i, y_i denote the horizontal and vertical position of landmark i, respectively. Observe that when a head rotates sideways, the horizontal distance between the tip of the nose and the corner of the eyes changes. To scale this measure properly, this distance is compared with the horizontal distance between both lateral eye corners. Thus, the fraction

$$\frac{|x_{\text{right lateral eye corner}} - x_{\text{nose tip}}|}{|x_{\text{right lateral eye corner}} - x_{\text{left lateral eye corner}}|} \tag{1}$$

is measured to approximate horizontal rotation (see Fig. 5). When a head is straight, the tip of the nose is assumed to be in the middle. But, when it rotates to a side, the fraction becomes smaller or larger, depending on the side it is rotating towards. Note that this fraction could be used to approximate the horizontal rotation in degrees using known facial rotations. However, varying nose shapes and facial asymmetries could influence the results.

Using a similar key insight, vertical rotation can be measured by comparing the vertical distance between the nose root and nose tip and the vertical distance between the nose root and the chin. Thus, the fraction

$$\frac{|y_{\text{nose root}} - y_{\text{nose tip}}|}{|y_{\text{nose root}} - y_{\text{chin}}|} \tag{2}$$

is measured to approximate the vertical rotation, see Fig. 6. Note that this measure is more subjective to personal traits, as nose lengths can vary. Although this may raise issues for an individual image, we believe that this method is sufficient for comparing the datasets generally, as individual errors will not have a large impact on the general comparison.

3.2. *Facial embedding with face recognition models*

Are new individuals created or are the generated images too similar to individuals from the input dataset? To compare the images from a human perspective, some kind of *facial embedding* is necessary. It is imperative that the dimensionality of each image is reduced. Every image consists of 1024 × 1024 pixels and each pixel consists of three color values (RGB). Given the size of these datasets, it is unfeasible to compare the pixels for each pair of images. Furthermore, a human does not compare two images pixel by pixel. Instead, one matches facial features such as eyes, nose, hair, and mouth to evaluate if these two images belong to the same person. This is why we decide to use face recognition models, where a face is first embedded to a point in a latent space, such that distances can be measured between faces. If two points are close, they are assumed to be similar. In this way, we can determine if new individuals are created. The four facial embedding methods used for facial recognition are outlined below.

***FaceNet*.** Schroff, Kalenichenko, and Philbin (2015) is well suited for our objective. It is a deep convolutional network that converts an image (160 × 160 pixels) to a 128-dimensional vector that lies on the 128-dimensional hypersphere. To find an appropriate embedding, *FaceNet* uses a triplet loss function.

***OpenFace*.** Amos, Ludwiczuk, and Satyanarayanan (2016) follows the same concept as FaceNet (Schroff et al., 2015). It is, however, open-source and focuses on real-time face recognition. It converts an image (96 × 96 pixels) to a 128-dimensional vector that lies on the 128-dimensional hypersphere.

***DeepFace*.** Taigman, Yang, Ranzato, and Wolf (2014) uses 3D face modeling and a large deep neural network to recognize faces. It converts an image (152 × 152 pixels) to a 4096-dimensional vector, which is then used to identify individuals using a classification layer. Taigman et al. (2014) call this vector the "raw face representation feature vector".

***VGG-face*.** Parkhi et al. (2015) uses the well-known VGG-16 architecture (Simonyan & Zisserman, 2014) to specifically train for facial recognition. It converts an image (224 × 224 pixels) to a 2622-dimensional vector. This model also uses the specific loss function from FaceNet (Schroff et al., 2015) to train the model for facial recognition.

Dimensionality reduction

The output vector of DeepFace (Taigman et al., 2014) and VGG-Face (Parkhi et al., 2015) is too large to properly cluster on. Therefore, the dimensionality is reduced with *singular value decomposition* (SVD) from a 4096- and 2622-dimensional vector to a 128-dimensional vector. Thus enforcing the same dimensions of the output vector for each embedding method. This dimensionality reduction could weaken the accuracy of these models, as some information is lost. However, if there is still a clear distinction between the datasets in this lower dimension, there must be a similar or larger, difference in the higher dimension. We call these models *Reduced DeepFace* and *Reduced VGG-Face* from now on.

3.3. Clustering

Once the faces are embedded using face recognition models, images can be compared. There are many options, however we will show why clustering is the most natural approach in our view. In the end, we want to answer the question if actual new persons are generated. The face recognition methods enable us to measure the distance between each pair of images. If the distance between images A and B is below a defined threshold, the images are considered to be of the same identical person. However, if the distance between images B and C is also below the threshold, images A, B and C all belong to the same person and form a cluster. Thus, a clustering approach naturally arises by this logic. Each cluster of images represents a single person, according to the face recognition methods.

To investigate if the output dataset contains the same identities as the input dataset, two combinations are made:

- **FFHQ** ∪ **($\boldsymbol{\psi = 1}$):** The input dataset combined with the generated images without truncation;
- **FFHQ** ∪ **($\boldsymbol{\psi = 0.5}$):** The input dataset combined with the generated images with truncation.

The clustering is done on the embeddings of these two combinations. Due to the size of the datasets (170,000 images in total), a clustering method with few parameters is preferred. Furthermore, there is no or little domain knowledge of proper parameter values, making most clustering methods too computationally expensive, as a range of values for the parameters needs to be evaluated.

This leads to the decision to use *HDBSCAN* (Campello, Moulavi, Zimek, & Sander, 2015). The idea behind this algorithm is that instances A and B are *neighbors* if the distance between them is less than or equal to ϵ and two instances A and B are in the same cluster if there exists a sequence of instances from A to B such that each successive instance is a neighbor of the previous. *HDBSCAN* allows ϵ to be altered post-completion. In this research, we use the implementation of McInnes, Healy, and Astels (2017) with the *Euclidean* distance function. Although *HDBSCAN* has computational complexity $\mathcal{O}(n^2)$ (Campello et al., 2015) with n the number of samples, McInnes, Healy, and Astels (2016) show that *HDBSCAN* performs reasonably fast for large datasets. Furthermore, it returns a hierarchical clustering. This is useful to determine different statistics post-completion. If instead the very similar *DBSCAN* (Ester, Kriegel, Sander, & Xu, 1996) is used, some information about the parameter ϵ is necessary. ϵ determines the neighborhood of each point. The relevant range for ϵ varies greatly for different embeddings. Without large computational costs, it is possible to determine the results for different values of ϵ using the hierarchical cluster, after running *HDBSCAN*.

HDBSCAN has a single primary input parameter m_{pts} (Campello et al., 2015). This parameter determines if a group of samples is large enough to be considered an actual cluster. If two images are embedded closely together, they should be able to form a cluster, as they can belong to the same person. Thus, $m_{pts} = 2$ is a natural choice, as it allows all cluster sizes except a cluster containing a single image.

3.3.1. Cluster evaluation

The goal of clustering is to investigate if the input and output datasets consist of different individuals. Therefore, *purity* (Manning, Raghavan, & Schütze, 2008) is used to measure the intertwinedness of the clustering, as this metric evaluates if subclusters consist of purely real or generated images. *Purity* is measured by counting the samples of the most frequent class in each cluster and dividing by the total number of samples. More formally, let clustering C of N samples consist of subclusters C_i for $i \in \{1, \ldots, K\}$, for some $K \in \mathbb{N}_{>0}$. Each sample j comes from a corresponding dataset labeled l_j. For each subcluster C_i, let d_i denote the label of the dataset that occurs most frequently, then:

$$\text{Purity}(C) = \frac{1}{N} \cdot \sum_{i=1}^{K} \sum_{j \in C_i} \mathbb{1}_{l_j = d_i}. \tag{3}$$

If $\text{Purity}(C) = 1$, it means that every subcluster only contains samples of one class. If there are only two classes, a lower bound of *purity* is $\text{Purity}(C) = 0.5$, as in the worst case every subcluster is split 50/50 between the classes.

Baseline purity. Note that the upper and lower bound, previously given, cannot always be achieved. This is dependent on the distribution of the labels and the structure of a clustering. For example, if there is only one cluster and the labels are divided 80/20, the purity score will be 0.8. Therefore, a better baseline is necessary to evaluate how good/bad a purity score of a clustering is.

Assume that the input and output dataset are sampled from the same distribution. Then there is no way of telling which image is drawn from which dataset. For each parameter combination, *HDBSCAN* returns a cluster with a certain structure. Each cluster consists of a number of subclusters all with a corresponding size. If there would be no difference between the two datasets, it would correspond with randomly assigning each sample to a position in the cluster. As we have seen before, the structure of the cluster is important for the purity score.

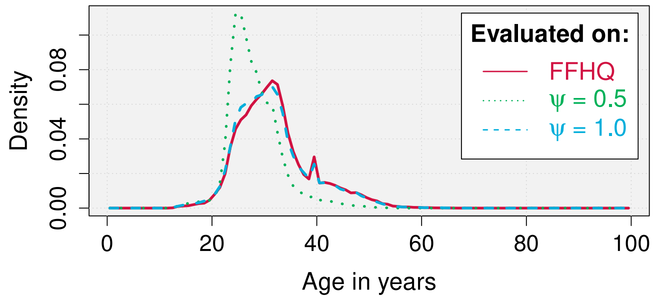

Fig. 7. Age distribution: Age distribution averaged per dataset.

Therefore, we approximate the expected purity score of a randomly assigned cluster with the same structure as provided by *HDBSCAN*. Under the hypothesis that there is no difference between the datasets, we get an average purity score that is ultimately used to compare the results. If the results are close to this baseline, it means that the datasets are very similar. On the other hand, if there is a clear distinction between the baseline and the results, it would mean that the datasets are not alike.

4. Analysis

The images from the datasets are analyzed in two ways. First, models are used to predict certain attributes of each face (e.g., gender and age). This will determine the distribution of these features, which can be used to compare the datasets globally. Second, multiple embedding methods are used in combination with a clustering method. By looking inside each subcluster and evaluating the purity score (see Section 3.3.1), a comparison between the datasets can be made. The results of both approaches are discussed below.

4.1. Results attributes

For each image in every dataset, models were used to predict the following attributes: *age*, *gender*, *race*, *horizontal rotation*, and *vertical rotation*. The results are grouped together per dataset. This gives a global overview of these attributes for each dataset. In particular, we are interested in the similarity of the distributions. If these distributions are different, it would suggest that the underlying datasets are in fact different.

4.1.1. Results age

Without truncation ($\psi = 1$), the age distribution of the generated images is almost identical to the input dataset (*FFHQ*), see Fig. 7. Even the small peak around 40 years is similar for these two datasets. If truncation is added ($\psi = 0.5$), it can be observed that the distribution shifts more towards the younger age groups.

4.1.2. Results gender

The model returns for both classes (*woman* and *man*) a probability value. The *dominant gender* is the gender with the highest probability of the two. Note that without truncation ($\psi = 1$), the distribution is almost the same as the input dataset (*FFHQ*), see Fig. 8. Whereas with truncation ($\psi = 0.5$), relatively more females are generated compared to the input.

4.1.3. Results race

Table 1 shows the average probability mass for each race. Table 2 shows the distribution of the *dominant race*. This is the class that obtained the maximum probability given by the model. Without truncation ($\psi = 1$), the distribution is very similar to the input dataset (*FFHQ*). However, if truncation is added ($\psi = 0.5$), *white* is predicted more often.

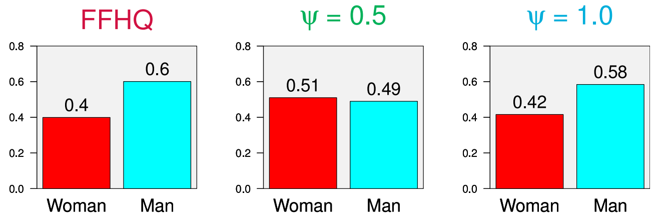

Fig. 8. Dominant gender: Probability averaged per dataset that man or woman gets the highest prediction probability.

Table 1
Average probability: Race distribution averaged per dataset.

Dataset	Race					
	Asian	Indian	Black	White	Middle Eastern	Latino Hispanic
FFHQ	0.1826	0.0425	0.0549	0.4889	0.0965	0.1346
$\psi = 0.5$	0.0921	0.0198	0.0118	0.6879	0.0805	0.1079
$\psi = 1$	0.1883	0.0366	0.0579	0.4900	0.0933	0.1339

Table 2
Dominant race probability: Probability averaged per dataset that a race class gets the highest prediction probability.

Dataset	Race					
	Asian	Indian	Black	White	Middle Eastern	Latino Hispanic
FFHQ	0.1961	0.0183	0.0509	0.5775	0.0513	0.1058
$\psi = 0.5$	0.1031	0.0034	0.0084	0.7769	0.0346	0.0735
$\psi = 1$	0.2066	0.0100	0.0552	0.5719	0.0501	0.1062

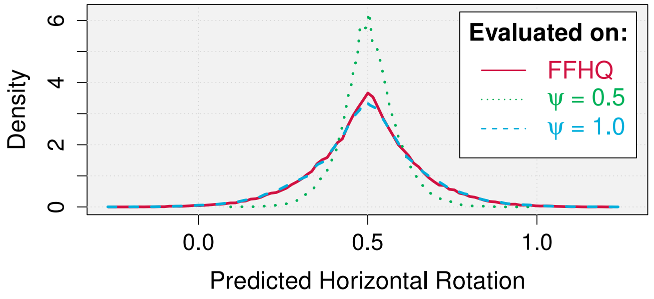

Fig. 9. Horizontal rotation: Horizontal rotation averaged per dataset, predicted using the landmarks of dlib (see Fig. 5).

4.1.4. Results horizontal rotation

As explained by Fig. 5, the horizontal rotation is measured using the predicted landmarks of *dlib* (see Eq. (1)). In Fig. 9, it can be observed that without truncation ($\psi = 1$), the distribution is nearly identical. When truncation is added, the distribution narrows to 0.5, which means that more straight faces are generated or the faces are more symmetric.

4.1.5. Results vertical rotation

As explained by Fig. 6, the vertical rotation is measured using the predicted landmarks of *dlib* (see Eq. (2)). Again, the distribution without truncation ($\psi = 1$) is identical to the distribution of the input dataset *FFHQ* (see Fig. 10). If truncation is added ($\psi = 0.5$), the distribution shifts to the right. There are two possible explanations. First, it could mean that the generated images are rotated more downwards. Second, it is possible that the generated images have a longer nose. In Fig. 11, the distributions of the nose length can be found. There is a significant shift when truncation is added ($\psi = 0.5$). Thus, it can be concluded that the nose lengths are on average larger for $\psi = 0.5$.

4.1.6. Failed detections deepface

The models that predict the age, gender, and race were trained using a specific face detector. When the detector finds a face, it automatically

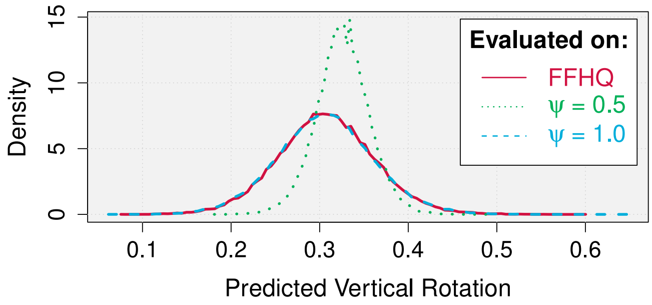

Fig. 10. Vertical rotation: Vertical rotation averaged per dataset, predicted using the landmarks of dlib (see Fig. 6).

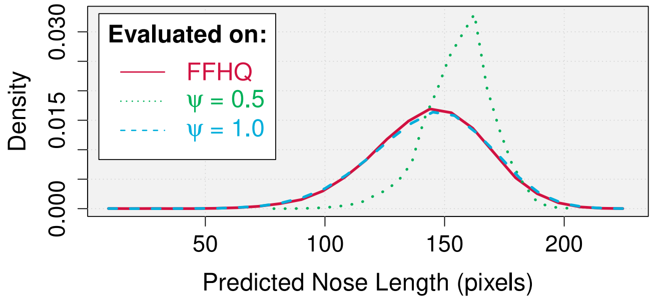

Fig. 11. Nose length: Nose length averaged per dataset, estimated by measuring the distance between the nose root and nose tip.

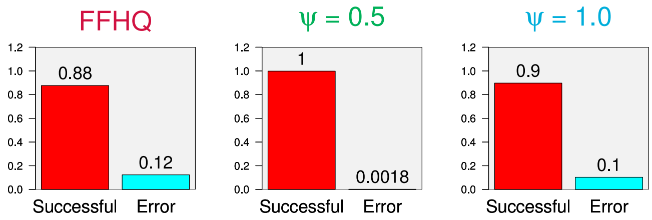

Fig. 12. Failure rate deepface detector: Probability averaged per dataset that the haarcascade detector (Bradski, 2000) (used in deepface) fails to detect a face.

trims and resizes the image. However, this detector sometimes fails to detect a face. In this case, the image is simply omitted from the attribute analysis. Fig. 12 shows how often the detector is successful. Note that with truncation ($\psi = 0.5$) this failure probability decreases drastically. The results for *FFHQ* and no truncation ($\psi = 1$) are similar and relatively high. A failure rate of around 10 percent is rather substantial.

In Figs. 13, 14, and 15, the first images of each dataset are shown where the *deepface* detector fails. Only for $\psi = 0.5$, it is not very clear why these images fail. However, we suspect that the following factors contribute to the general failure of the detector:

- eyewear;
- headwear;
- rotated heads;
- multiple persons;
- young age;
- obstructed eyes;
- structural errors (deformation, glitches, missing parts, etc.).

Note that these are only visual observations and should be investigated further.

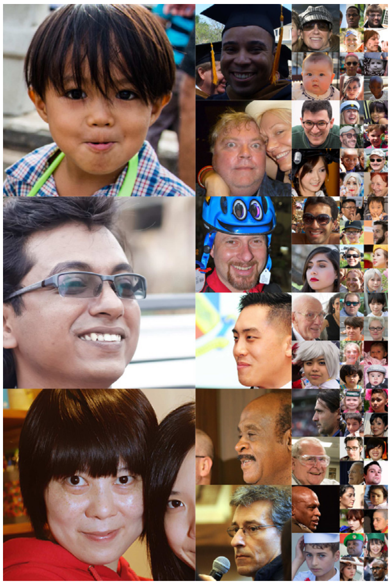

Fig. 13. Deepface detector failures FFHQ: The first images from the input dataset FFHQ, where the deepface detector does not detect a face.

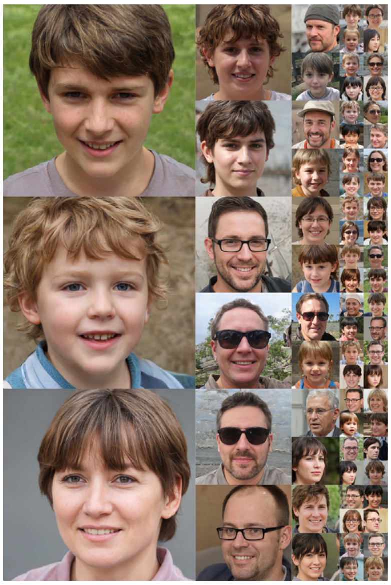

Fig. 14. Deepface detector failures $\psi = 0.5$: The first images from the output dataset with truncation ($\psi = 0.5$), where the deepface detector does not detect a face.

4.1.7. Failed detections dlib

Dlib uses another face detector. The failure rate of this detector is also measured. As can be seen in Fig. 17, the failure probability is much lower compared to Fig. 12. It is notable that if truncation is added ($\psi = 0.5$), the failure probability is even zero. However, the differences between the probabilities are so small that it is hard to draw any meaningful conclusions for the different datasets. The failure rate is very small for each dataset.

In Figs. 16 and 18, the first images of each dataset are shown, where the *dlib* detector fails. Note that for $\psi = 0.5$, there are no failures. The same elements we observed in the failures of the *deepface* detector are prevalent in the *dlib* detector failures. However, the *dlib* detector seems to be more robust compared to the *deepface* detector.

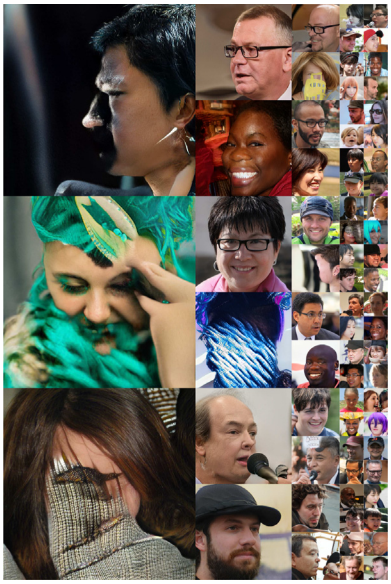

Fig. 15. Deepface detector failures $\psi = 1$: The first images from the output dataset without truncation ($\psi = 1$), where the deepface detector does not detect a face.

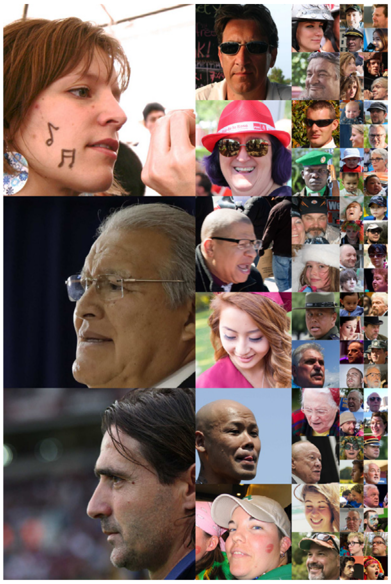

Fig. 16. Dlib detector failures FFHQ: The first images from the input dataset FFHQ, where the dlib detector does not detect a face.

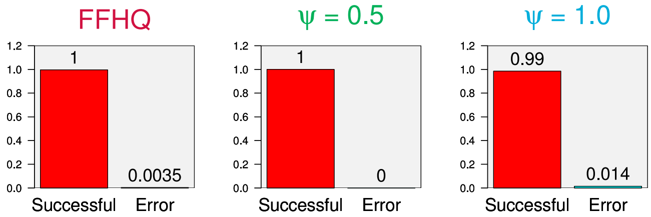

Fig. 17. Failure rate dlib detector: Probability averaged per dataset that the frontal face detector (used in dlib) fails to detect a face.

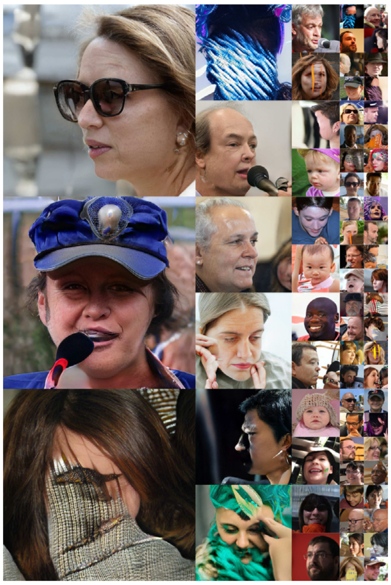

Fig. 18. Dlib detector failures $\psi = 1$: The first images from the output dataset without truncation ($\psi = 1$), where the dlib detector does not detect a face.

Table 3
Maximum subclusters: For each dataset combination and facial embedding method, the maximum number of subclusters is determined with $m_{pts} = 2$.

Facial embedding	Dataset combination	
	FFHQ ∪ ($\psi = 1$)	FFHQ ∪ ($\psi = 0.5$)
FaceNet	5170	5592
OpenFace	2835	3598
Reduced DeepFace	2885	3153
Reduced VGG-Face	2865	2246

4.2. Results clustering

In Section 3.3, it is discussed why clustering is a natural approach to determine if the newly generated images belong to an existing person. The clustering results can be seen in Fig. 19. Note that different parameter values of ϵ are relevant for each embedding method. This makes *HDBSCAN* (Campello et al., 2015) very useful, as the parameter value of ϵ can be changed post-computation. Given the parameters, *HDBSCAN* returns a cluster. Two measures are of our interest. First, the number of subclusters within each cluster. This indicates how many unique persons exist in the data according to the embedding methods. Second, the *purity* of a cluster is measured (see Section 3.3.1). We cluster on the combination of the input dataset (*FFHQ*) and the output dataset with either no truncation ($\psi = 1$) or with truncation ($\psi = 0.5$). In this way, the output dataset can be compared with the input dataset.

For each facial embedding the maximum number of subclusters (Table 3) is determined with $m_{pts} = 2$ (see Section 3.3).

4.2.1. Purity results

The purity results for $m_{pts} = 2$ are shown in Fig. 19. The relevant range for ϵ is chosen based on the number of clusters. Two main conclusions can be drawn from these graphs. First of all, 7 out of 8 clusterings show a clear distinction between the baseline and the actual purity score. Only OpenFace without truncation ($\psi = 1$) shows no obvious separation. Therefore, it can be concluded that there is a definite difference between the input and the output datasets. Thus, the generated images belong to different persons compared to the input dataset, according to the facial recognition methods. Second, the gap between the baseline and the actual purity score is much larger with truncation ($\psi = 0.5$) than without truncation ($\psi = 1.0$). Thus, truncation makes it more likely that a cluster is predominantly real or generated.

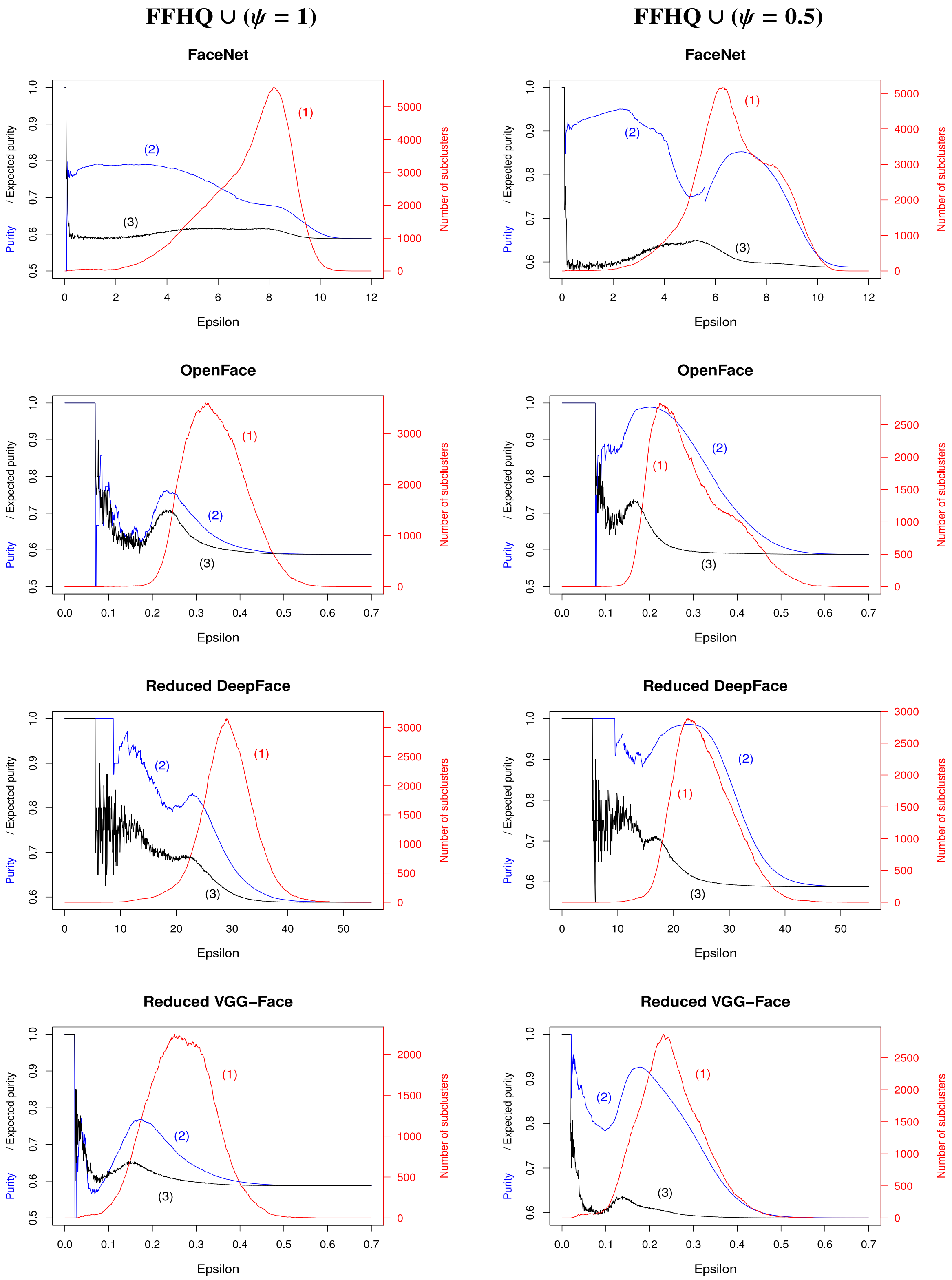

Fig. 19. Clustering purity: Using HDBSCAN with $m_{pts} = 2$, ϵ determines which clustering is made. The red line (1) denotes the number of subclusters of each cluster. The blue line (2) is the purity score (see Section 3.3.1). The black line (3) shows the approximated purity score under the hypothesis that the two datasets are similarly distributed using 100 simulations (see Section 3.3.1). The left side is the combination FFHQ ∪ ($\psi = 1$), whereas the right side is the combination FFHQ ∪ ($\psi = 0.5$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5. Conclusion and further research

We presented a general two-pronged approach that tries to humanly compare the input and output datasets for a given face generator. However, we explicitly applied this approach to the state-of-the-art generator StyleGAN2 (Karras et al., 2019). We started by comparing the input dataset (*FFHQ*) and the output datasets based on their attributes. Multiple models were used to predict attributes (*age*, *gender*, *race*, *horizontal*, and *vertical rotation*) for each image. The results were very clear. The attribute distributions were the same for the input dataset and the generated images without truncation ($\psi = 1$). However, when truncation is added ($\psi = 0.5$), the attribute distributions shift significantly towards the attributes corresponding with the latent variable used in truncation. Although there exist many evaluation measures for GANs (Borji, 2018), the three most commonly used measures are: *Fréchet Inception Distance* (FID), *Inception Score* (IS), and *Precision and Recall* (P&R) (Borji, 2021; Shmelkov, Schmid, & Alahari, 2018). *FID* measures the difference between the input and output images by embedding them into the feature space of an Inception Net (trained on ImageNet) (Borji, 2018). *IS* also uses the Inception Net to measure the diversity of the generated images compared to the mean. *P&R* quantifies how similar the generated images are to the input dataset and how well the entire training dataset is covered. Additionally, StyleGAN2 (Karras et al., 2019) evaluates the *perceptual path length*, (PPL) which measures the difference between the VGG-16 embedding (Simonyan & Zisserman, 2014) of two consecutive images, where a path in the latent space is subdivided into linear segments (Karras et al., 2018). These measures have been used to evaluate the performance of StyleGAN2 (Karras et al., 2019). Thus, the observation that StyleGAN2 is able to learn the input dataset is not new. It is known that GANs are able to learn the input distribution, although training sometimes appears successful, whilst the target distribution is actually far from the trained distribution (Arora, Ge, Liang, Ma, & Zhang, 2017). However, it has not previously been shown that higher-level human concepts are also preserved. It could be that somehow these measures indirectly assess these human concepts, although this has not yet been show. This article gives a direct approach and demonstrates that such human concepts are indeed preserved, which further strengthens the work of Karras et al. (2019).

In addition, four facial embedding models (*FaceNet*, *OpenFace*, *Reduced DeepFace*, and *Reduced VGG-Face*) were used to embed the images. This allowed us to cluster each combination of input and output dataset. By determining the purity score, which measures how intertwined each subcluster is, we were able to show that the generated images are not grouped together with the input dataset. This means that StyleGAN2 is able to generate new persons that do not exist in the input dataset, according to the facial embeddings. Recently, Khodadadeh et al. (2022) had a similar idea of using a face recognition method in combination with StyleGAN2. They used *FaceNet* in a loss function to generate faces with StyleGAN2 that belong to the same identity. Furthermore, they used 35 attribute methods to steer the latent space in order to generate faces with modified attributes, which is different compared to our research. The insight that StyleGAN2 is capable of generating new identities is novel and one of the contributions of our research.

Summarizing, by using a two-pronged humanly approach, consisting of predicting human attributes (Section 3.1) and clustering using face recognition models (Sections 3.2 and 3.3), the following conclusions can be drawn:

- The images generated by StyleGAN2 (without truncation) have the same attribute distributions as the input dataset, according to the prediction models.
- The generated faces belong with high probability to different persons compared to the input dataset, according to the clustering using face recognition models.
- Adding truncation to the latent variable space changes the attribute distributions towards the attributes corresponding with the latent variable used in truncation, according to the prediction models.

Generalizing, our approach can also be used for other face generators. It is not specifically tailored for StyleGAN2. Furthermore, our approach is modular in the sense that different attribute prediction and facial embedding methods can be added or removed. It should therefore be used in conjunction with other evaluation measures such as *FID* and *PPL* to give a broader perspective of the performance of a face generator. It addresses different questions and concerns compared to previous measures. When our approach, for example, shows that the generated images belong to identities in the input dataset, adaptations could be made if this effect is undesirable due to privacy issues.

5.1. Future work

Finally, we address a number of topics for future research. Section 4 provides multiple insights that should be explored further. First, note that the maximum number of subclusters is relatively small (see Table 3). Not more than 5592 subclusters are formed maximally for a dataset consisting of 170,000 images. A lot of images are considered to be anomalies, which means that there is no face that closely resembles theirs. It could be that either the dataset is too small, due to the wide variety of possible faces, or the embedding methods are too specific.

Second, truncation ensures that the latent variables lie closer to the expected intermediate latent variable $\bar{\boldsymbol{w}}$ (see Section 2.1). In the results, the attributes were very similar for the input dataset and the output dataset without truncation ($\psi = 1$). However, when truncation is added ($\psi = 0.5$), there was a shift in the attribute distributions. Our hypothesis is that this shift stems from the attribute values of the images generated with $\bar{\boldsymbol{w}}$ as intermediate latent variable (see Fig. 2). Taking the average of the predicted attribute values for the first 1000 images, generated with $\bar{\boldsymbol{w}}$, leads to Table 4.

When we examine the differences in the attribute distributions between truncation ($\psi = 0.5$) and no truncation ($\psi = 1$), and compare these with the difference between Tables 4 and 5, we see that they coincide. Thus, we hypothesize that adding truncation focuses the attribute distributions towards the attribute values belonging to the faces generated with the expected intermediate latent variable $\bar{\boldsymbol{w}}$. Karras et al. (2019) regulate the generator to smoothen the perceptual path length of generated images under small perturbations in the latent space. This could be a reason why the human attributes are also similar under small perturbations. The hypothesis can be tested by replacing $\bar{\boldsymbol{w}}$ in the truncation procedure by a different intermediate latent variable and investigating the attribute distributions of the newly generated images. If the hypothesis holds, this method can also be used to generate images with the desired attributes. Future research is needed to explore how $\bar{\boldsymbol{w}}$ and the truncation variable influences the attribute distributions. The attribution methods that were used are all trained on alternate datasets. It is unclear how well their performance transfers to the data that is used in this research. Nevertheless, it still provides the insight that the input and output distributions were similar. Still, it remains interesting to evaluate how well the models can transfer their learned knowledge to this dataset. Additionally, the goal of our approach was to evaluate the generator in a more humanlike way. We decided to use methods that were trained using humanly labeled data, as it was unfeasible for us to label this dataset ourselves. However, it remains uncertain how 'humanlike' these methods are. Are they actually predicting correct attribute labels for our datasets? Although this questions is beyond our scope, it is interesting to evaluate if these models are good at replacing human experts. Lastly, the facial recognition methods are trained using datasets of real images. The results showed that the generated images are embedded differently than the input images. If the facial embedding methods are also trained using generated faces, a better comparison could possibly be made.

CRediT authorship contribution statement

Joris Pries: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Sandjai Bhulai:** Conceptualization, Validation, Writing – review & editing, Supervision. **Rob van der Mei:** Conceptualization, Validation, Writing – review & editing, Supervision.

Table 4
Attribute predictions $\bar{w}$: Average predicted attribute values for 1000 images (seeds 0000–0999) generated with the expected intermediate latent variable $\bar{\boldsymbol{w}}$.

Age (years)	Gender (prob.)		Race (prob.)		Horizontal rotation	Vertical rotation	Failed deepface (prob.)	Failed dlib (prob.)
23.7	Woman 0.649	Man 0.351	Asian	Indian	0.504	0.345	0	0
			0.0009	0.0002				
			Black	White				
			0.0002	0.9975				
			Middle eastern	Latino Hispanic				
			0.0006	0.0005				

Table 5
Attribute predictions ($\psi = 1$): Averaged attribute values of the output dataset without truncation.

Age (years)	Gender (prob.)		Race (prob.)		Horizontal rotation	Vertical rotation	Failed deepface (prob.)	Failed dlib (prob.)
31.6	Woman 0.4158	Man 0.5842	Asian	Indian	0.497	0.306	0.1	0.014
			0.2066	0.0100				
			Black	White				
			0.0552	0.5719				
			Middle eastern	Latino Hispanic				
			0.0501	0.1062				

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and material

All data used in this research is cited in the appropriate sections.

Acknowledgments

The authors wish thank the anonymous referees for their useful comments, which has led to a significant improvement of the readability and quality of the paper.

References

Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). *OpenFace: A general-purpose face recognition library with mobile applications: Technical report. CMU-CS-16-118*, CMU School of Computer Science.

Arora, S., Ge, R., Liang, Y., Ma, T., & Zhang, Y. (2017). Generalization and equilibrium in generative adversarial nets (gans). CoRR abs/1703.00573 URL: http://arxiv.org/abs/1703.00573 arXiv:1703.00573.

Borji, A. (2018). Pros and cons of gan evaluation measures. http://dx.doi.org/10.48550/ARXIV.1802.03446, URL: https://arxiv.org/abs/1802.03446.

Borji, A. (2021). Pros and cons of gan evaluation measures: New developments. http://dx.doi.org/10.48550/ARXIV.2103.09396, URL: https://arxiv.org/abs/2103.09396.

Bradski, G. (2000). The opencv library. *Dr. Dobb's Journal of Software Tools*.

Breitenstein, M. D., Kuettel, D., Weise, T., van Gool, L., & Pfister, H. (2008). Real-time face pose estimation from single range images. In *2008 IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale gan training for high fidelity natural image synthesis. arXiv:1809.11096.

Buolamwini, J., & Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In S. A. Friedler, & C. Wilson (Eds.), *Proceedings of the 1st conference on fairness, accountability and transparency* (pp. 77–91). New York, NY, USA: PMLR, URL: http://proceedings.mlr.press/v81/buolamwini18a.html.

Campello, R. J. G. B., Moulavi, D., Zimek, A., & Sander, J. (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data*, 10. http://dx.doi.org/10.1145/2733381.

Díaz Barros, J. M., Mirbach, B., Garcia, F., Varanasi, K., & Stricker, D. (2019). Real-time head pose estimation by tracking and detection of keypoints and facial landmarks. In D. Bechmann, M. Chessa, A. P. Cláudio, F. Imai, A. Kerren, P. Richard, A. Telea, & A. Tremeau (Eds.), *Computer vision, imaging and computer graphics theory and applications* (pp. 326–349). Cham: Springer International Publishing.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise* (pp. 226–231). AAAI Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), *Advances in neural information processing systems, Vol. 27* (pp. 2672–2680). Curran Associates, Inc., URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a nash equilibrium. CoRR abs/1706.08500 URL: http://arxiv.org/abs/1706.08500 arXiv:1706.08500.

Hu, Yuxiao, Chen, Longbin, Zhou, Yi, & Zhang, Hongjiang (2004). Estimating face pose by facial asymmetry and geometry. In *Sixth IEEE international conference on automatic face and gesture recognition, 2004. proceedings* (pp. 651–656).

Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. CoRR abs/1710.10196 URL: http://arxiv.org/abs/1710.10196 arXiv:1710.10196.

Karras, T., Laine, S., & Aila, T. (2018). A style-based generator architecture for generative adversarial networks. CoRR abs/1812.04948 URL: http://arxiv.org/abs/1812.04948 arXiv:1812.04948.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2019). Analyzing and improving the image quality of stylegan. CoRR abs/1912.04958.

Khodadadeh, S., Ghadar, S., Motiian, S., Lin, W. A., Bölöni, L., & Kalarot, R. 2022. Latent to latent: A learned mapper for identity preserving editing of multiple face attributes in stylegan-generated images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV)* (pp. 3184–3192).

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

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge, UK: Cambridge University Press, URL: http://nlp.stanford.edu/IR-book/information-retrieval-book.html.

McInnes, L., Healy, J., & Astels, S. (2016). Benchmarking performance and scaling of python clustering algorithms. URL: https://hdbscan.readthedocs.io/en/latest/performance_and_scalability.html.

McInnes, L., Healy, J., & Astels, S. (2017). Hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2. http://dx.doi.org/10.21105/joss.00205.

Merler, M., Ratha, N. K., Feris, R. S., & Smith, J. R. (2019). Diversity in faces. CoRR abs/1901.10436 URL: http://arxiv.org/abs/1901.10436 arXiv:1901.10436.

Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. In M. W. J. Xianghua Xie, & G. K. L. Tam (Eds.), *Proceedings of the British machine vision conference* (pp. 41.1–41.12). BMVA Press, http://dx.doi.org/10.5244/C.29.41.

Rothe, R., Timofte, R., & Van Gool, L. (2018). Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, *126*, 144–157. http://dx.doi.org/10.1007/s11263-016-0940-3.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). 300 Faces in-the-wild challenge: The first facial landmark localization challenge. In *2013 IEEE international conference on computer vision workshops* (pp. 397–403).

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., et al. (2016). Improved techniques for training gans. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc, URL: https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. CoRR abs/1503.03832 URL: http://arxiv.org/abs/1503.03832 arXiv:1503.03832.

Serengil, S. I., & Ozpinar, A. (2020). Lightface: A hybrid deep face recognition framework. In *2020 innovations in intelligent systems and applications conference* (pp. 23–27). IEEE, http://dx.doi.org/10.1109/ASYU50717.2020.9259802.

Shen, Y., Yang, C., Tang, X., & Zhou, B. (2020). Interfacegan: Interpreting the disentangled face representation learned by gans. arXiv:2005.09635.

Shmelkov, K., Schmid, C., & Alahari, K. (2018). How good is my gan?. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer vision – ECCV 2018* (pp. 218–234). Cham: Springer International Publishing.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE conference on computer vision and pattern recognition* (pp. 1701–1708).

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., et al. (2016). Yfcc100m. *Communications of the ACM*, *59*, 64–73. http://dx.doi.org/10.1145/2812802.

Vezhnevets, V., Sazonov, V., & Andreeva, A. (2003). A survey on pixel-based skin color detection techniques. In *IN PROC. GRAPHICON-2003* (pp. 85–92).

West, J., & Bergstrom, C. (2019). Which face is real?. URL: http://www.whichfaceisreal.com/learn.html.

Xu, J., Jin, L., Liang, L., Feng, Z., Xie, D., & Mao, H. (2017). Facial attractiveness prediction using psychologically inspired convolutional neural network (pi-cnn). In *2017 IEEE international conference on acoustics, speech and signal processing* (pp. 1657–1661).

Zhou, S., Gordon, M. L., Krishna, R., Narcomey, A., Fei-Fei, L., & Bernstein, M. S. (2019). Hype: A benchmark for human eye perceptual evaluation of generative models. http://dx.doi.org/10.48550/ARXIV.1904.01121, URL: https://arxiv.org/abs/1904.01121.