applied sciences

MDPI

Article

Identifying Synthetic Faces through GAN Inversion and Biometric Traits Analysis

Cecilia Pasquini [1,2,*], Francesco Laiti [3,†], Davide Lobba [3,†], Giovanni Ambrosi [3], Giulia Boato [3,4] and Francesco De Natale [3,4]

[1] Fondazione Bruno Kessler, 38123 Povo, Italy
[2] Futuro e Conoscenza srl, 00138 Rome, Italy
[3] Department of Information Engineering and Computer Science, University of Trento, 38122 Trento, Italy
[4] CNIT—Consorzio Nazionale Interuniversitario per le Telecomunicazioni, 43124 Parma, Italy
* Correspondence: cecilia.pasquini@unitn.it
† These authors contributed equally to this work.

Abstract: In the field of image forensics, notable attention has been recently paid toward the detection of synthetic contents created through Generative Adversarial Networks (GANs), especially face images. This work explores a classification methodology inspired by the inner architecture of typical GANs, where vectors in a low-dimensional latent space are transformed by the generator into meaningful high-dimensional images. In particular, the proposed detector exploits the inversion of the GAN synthesis process: given a face image under investigation, we identify the point in the GAN latent space which more closely reconstructs it; we project the vector back into the image space, and we compare the resulting image with the actual one. Through experimental tests on widely known datasets (including FFHQ, CelebA, LFW, and Caltech), we demonstrate that real faces can be accurately discriminated from GAN-generated ones by properly capturing the facial traits through different feature representations. In particular, features based on facial landmarks fed to a Support Vector Machine consistently yield a global accuracy of above 88% for each dataset. Furthermore, we experimentally prove that the proposed detector is robust concerning routinely applied post-processing operations.

Keywords: Generative Adversarial Networks; image forensics; GAN inversion; face biometrics; StyleGAN2

check for updates

Citation: Pasquini, C.; Laiti, F.; Lobba, D.; Ambrosi, G.; Boato G.; De Natale, F. Identifying Synthetic Faces through GAN Inversion and Biometric Traits Analysis. *Appl. Sci.* **2023**, *13*, 816. https://doi.org/10.3390/app13020816

Academic Editors: Jana Dittmann and Andrey Makrushin

Received: 29 November 2022
Revised: 29 December 2022
Accepted: 31 December 2022
Published: 6 January 2023

1. Introduction

The creation of synthetic media through artificial intelligence has reached unprecedented levels of realism. Impressive results have been achieved in recent years for the semantic generation and manipulation of audio-visual content in fully or semi-automated fashion, also with multi-domain capabilities (e.g., text-to-image (https://openai.com/dall-e-2/ (accessed on 30 December 2022), text-to-speech).

Great effort has been spent in the generation of synthetic visual data. Video signals are highly powerful carriers in terms of semantics that can be conveyed, but they are still more complex to synthesize and manipulate. In fact, despite the rapidly progressing technology, producing high-quality manipulated videos involving arbitrary subjects and scenes still requires considerable skills and processing time. Instead, the generation of still pictures, and in particular the production of synthetic faces, is currently very easy and accessible to everyone, especially thanks to Generative Adversarial Networks (GANs), which can achieve impressive visual quality with minimal computational requirements [1,2]. Generative models are available online [3,4] to automatically generate or even edit face images; web interfaces (https://thispersondoesnotexist.com/ (accessed on 30 December 2022)) running pre-trained generators are also available, requiring no more than a click to obtain a hyper-realistic fake face. Harmful misuses of this technology have already been observed in the web ecosystem, including the creation of fictitious social media profiles (https://edition.cnn.com/2020/02/28/tech/fake-twitter-candidate-2020/index.html (accessed on 30 December 2022)) and digital identities (https://www.nbcnews.com/tech/security/how-fake-persona-laid-groundwork-hunter-biden-conspiracy-deluge-n1245387 (accessed on 30 December 2022)), thus calling for specialized detection technologies.

Researchers have proposed different techniques to distinguish between real and synthetic faces over the years [5–7], and great attention has been devoted over the last few years toward identifying whether an image has been generated through a GAN [8–11]. Further details about the related works are provided in the next section.

In this paper, we propose a novel detector for GAN-generated images based on the analysis of an image resulting from a *GAN inversion process*. In practice, given an image under investigation, we project it in the GAN latent space through an inversion process [12], and then back into the image space with a generation process. The resulting image is then compared with the actual one using different similarity metrics. We demonstrate that, as expected, when the process is applied to GAN-generated images, the two images will be extremely close to each other. In contrast, natural images will be approximated with significantly lower accuracy.

Extensive experiments on images coming from different sources have shown that landmark-based metrics are particularly effective in capturing the distinctive traits of synthetic images, which can be learned using shallow classifiers such as SVMs. Furthermore, the obtained detectors are proven to be generally robust to typical post-processing, such as resizing, JPEG compression, and upload/download operations through social media.

The major contributions of the paper can be summarized in the following points:

- We explicitly use the underlying mechanisms of GAN generators to perform the detection, instead of applying a blind learning procedure;
- We demonstrate that generative approaches produce structural errors in the reproduction of previously unseen face images, which can be revealed through appropriate sets of features;
- The proposed technique can be extended to any generator that admits an inversion, thus limiting the need for retraining over large image datasets.
- We release a data corpus of face images and their reconstructions through the StyleGAN2 inversion, available for research purposes.

The rest of the paper is structured as follows: in Section 2, we summarize the current state of research in the field; in Section 3, we describe the proposed inversion-based detector, providing details on the inversion process, and on the feature extraction and classification process; in Section 4, we define the experimental setup and the datasets used for testing, and we analyze the results under different operating conditions; finally, in Section 5, we draw some conclusions.

2. Related Work

In this section, we provide a short survey of the literature on real-versus-generated image detection, focusing in particular on data-driven methods and on the problem of GAN generation and inversion.

2.1. Data-Driven Detection Methods

In the context of real-versus-generated image detection, several approaches have been proposed. As in many related fields, deep networks have been widely exploited for detection purposes. One possibility is to apply them to characterize handcrafted features, as it happens in [13] for co-occurrence matrices. In addition, their inner behavior regarding neuron activation can be used as a clue to detect anomalies due to a synthetic source [14].

However, the most common approach is to employ fully data-driven methods, typically based on Convolutional Neural Networks (CNN), where the most distinctive features [11] are automatically learned from the data with remarkable results. Although they achieve excellent performance under rather aligned conditions in terms of training and testing distributions, it has been shown that they suffer some shortcomings due to their purely inductive nature [8]. This includes a significant loss of performance when the investigated data have undergone some post-processing, perhaps unseen in training, which likely occurs during the digital image life cycle. Data augmentation strategies and the inclusion in the training sets of post-processed samples may help in mitigating these issues [8,15], but they require the simulation of a huge variety of processing pipelines arising in real-world scenarios. Furthermore, deep learning-based approaches are often used as black boxes, thus making it difficult to interpret the inner mechanisms that led to a given decision.

For the above reasons, the performance and reliability of purely data-driven GAN detectors on testing data from uncontrolled settings are hard to predict. On the contrary, principled approaches exploiting the inherent architectural properties of generators represent a promising path to enrich the forensic tools available, and for devising detectors with enhanced generalization and explainability. Accordingly, this work explores the possibility of exploiting the inversion properties of the generator for identifying synthesized data, as explained in the following.

2.2. GAN Image Generation and Inversion

The GANs' typical inner mechanism entails that (random) vectors lying in a latent space are transformed into semantically meaningful images. This procedure can be also inverted to some extent, by back-projecting an image into a point in the latent space that corresponds to similar content in the image space.

The inversion of generative models has recently drawn strong attention in the computer vision community [12,16]. In this context, an interesting property is that the latent space can be queried and browsed along specific directions corresponding to visual attributes. As a matter of fact, inversion processes are mainly investigated for fast image editing applications. To the best of our knowledge, the only work that exploits inversion properties for inferring information on the image source is [17]. However, in that case, the authors analyze synthetic images only, with the goal of identifying the correct generator among a set of candidates. Moreover, a single distance-based indicator is used, and earlier, less compelling GANs are considered in the experimental analysis.

In our work, we analyze the outcome of the GAN inversion process, which given an image under investigation, finds the point in the GAN latent space that leads to the closest possible generated output in the image domain [12]. We expect that the application of this kind of process to images synthesized by that generator will produce a point in the latent space that leads to an equal or highly similar image, given that such a point exists for sure. On the contrary, the inversion process applied to natural images can only provide a latent vector that is associated with some approximation of the image, according to the considered generative model.

The proposed detector was tested on the widely known and highly realistic StyleGAN2 face generator [4], among the best-performing GAN image synthesizers currently available. The images under investigation are compared to their closest reconstruction obtained through the inversion process available in StyleGAN2, and their biometric facial traits are encoded through different face representations (including deep embeddings and landmark-based features) and learned by conventional classifiers such as SVMs. Conceptually, our work shares similarities with differential morphing detection pipelines, as in [18,19], where face image pairs containing authentic or morphed faces in biometric verification scenarios need to be distinguished, thus also requiring the characterization of subtle differences in facial traits. The use of handcrafted features and lightweight classifiers for detecting synthetic images has also been explored in [20], which however, did not include semantic features. The use of semantic cues for the detection of GAN-generated images has been explored and advocated by several works [21–24] but, in those cases, semantic artifacts are characterized in a post hoc analysis, thus not relying on the architecture of the generator.

3. Inversion-Based Detection

The key idea of the devised forensic strategy is to discriminate between real and GAN-synthesized face images by retro-projecting the image under analysis into the GAN latent space, synthesizing the image associated with the relevant point in the latent space, and comparing the generated image with the actual one.

This concept is represented in Figure 1. We start from an input image under investigation $\mathbf{x}_\mathrm{I}$ (the picture on top of image space), and we feed it into a GAN inversion process to obtain the corresponding point $\mathbf{z}_\mathrm{R}$ in the GAN latent space. We then run the GAN generator to produce a reconstructed version $\mathbf{x}_\mathrm{R}$ of the original image, associated with $\mathbf{z}_\mathrm{R}$ (bottom image in the image space). At this point, we have two copies of the image under analysis, the original and the reconstructed one, and we want to compare them to measure their similarity: the greater the similarity, the higher the probability that the image is GAN-generated. In fact, if the image under analysis comes from this generator, independently of the fact that it was part of the training set or not, a point in the latent space should exist that closely generates the same image. This is not true in the case of a real image, for which that point, in general, will not exist, and the inversion will just provide a more or less accurate approximation but not a perfect reconstruction. Indeed, in practice, we will never obtain a pointwise-equal image due to the limited accuracy of the inversion process, but nevertheless, we expect that GAN-generated images will much more closely match the target than real images.

Thus, properly comparing $\mathbf{x}_\mathrm{I}$ and $\mathbf{x}_\mathrm{R}$ is an important part of the process, as the differences among the two images cannot be just modeled as random noise. For this reason, we jointly perform two types of analysis: one based on standard image similarity measures, and the other on more specific face similarity features, and we analyze the relevant performances.

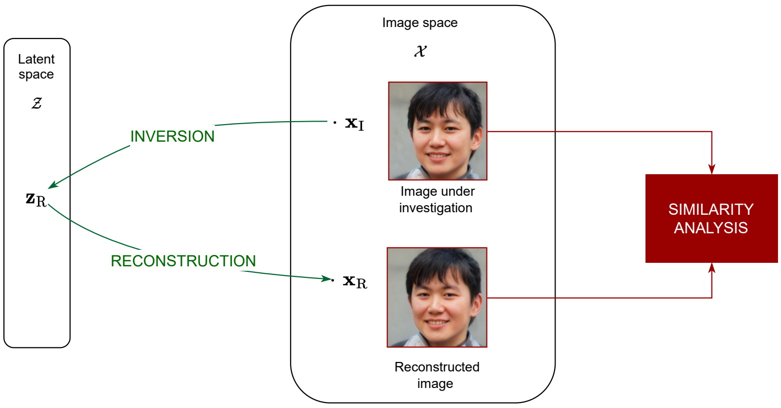


Figure 1. Overview of the inversion-based detection.

In the following sub-sections, we further illustrate the two main processes concerned with the above scheme: the inversion process, and the comparison and classification processes.

3.1. Inversion Process

Let us consider the typical GAN architecture, where a generator G and a discriminator D are trained jointly through an adversarial process. The goal of G is to generate synthetic data that resemble real data; the goal of D is to correctly distinguish the synthetic data generated by G and an available corpus of real data [25]. Starting from an initial version of both, the two networks are trained in an alternate manner by competing with each other and progressively improving their performance: the current version of D is fine-tuned on the real samples, and the synthetic ones created by the current version of G; in turn, G is then fine-tuned so as to maximize the classification loss of the updated version of D. At the end of this training process, the distribution of data generated by G is intended to match the distribution of real data, so as to minimize the discrimination capabilities of D.

Generators typically work to transform a randomly drawn vector $\mathbf{z} \in \mathcal{Z}$ into an image $\mathbf{x} = G(\mathbf{z})$ in the image space $\mathcal{X}$. $\mathbf{z}$ and $\mathcal{Z}$ represent the *latent vector* and the *latent space*, respectively, and $\mathbf{z}$ encodes information about the appearance of $G(\mathbf{z})$. Figure 2 depicts this mechanism. In other words, during the training process, the mapping $G : \mathcal{Z} \rightarrow \mathcal{X}$ is learned, and points that are close in the latent space $\mathcal{Z}$ are transformed through G into visually similar images in $\mathcal{X}$.

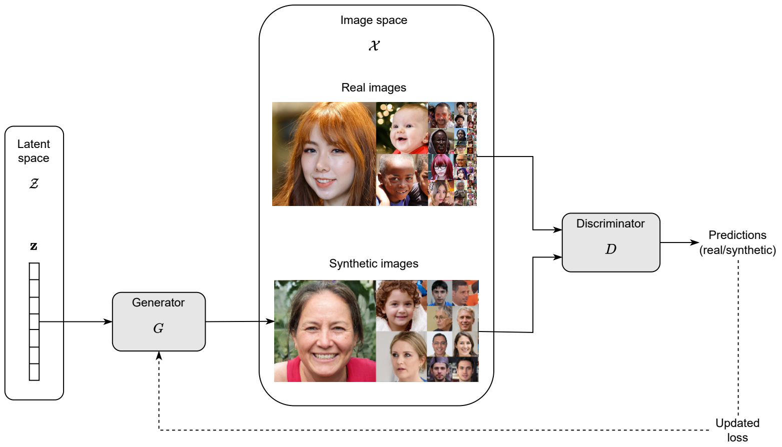


Figure 2. GAN architecture and training scheme.

The *inversion* (or *projection*) of GANs [12] consists of mapping a certain image under analysis $\mathbf{x}_\mathrm{I}$ back into its corresponding representation in the latent space. Formally, this can be formulated as finding a point $\mathbf{z}_\mathrm{R}$ in $\mathcal{Z}$ such that $G(\mathbf{z}_\mathrm{R})$ is as close as possible to $\mathbf{x}_\mathrm{I}$ according to a certain metrics $\ell(\cdot,\cdot)$:

$$\mathbf{z}_\mathrm{R} = \arg\min_{\mathbf{z}\in\mathcal{Z}} \ell(G(\mathbf{z}), \mathbf{x}_\mathrm{I}) \tag{1}$$

We denote $\mathbf{x}_\mathrm{R} \doteq G(\mathbf{z}_\mathrm{R})$ as the *reconstructed* version of $\mathbf{x}_\mathrm{I}$.

In this work, we study the inversion and reconstruction processes by focusing on the case where the generator G is the widely known StyleGAN2 face generator [3], for which the authors also provide a strategy for solving the problem in (1) (see Section 5 of [3]), together with an open source implementation (https://github.com/NVlabs/stylegan2-ada-pytorch (accessed on 30 December 2022)).

Figure 3 reports examples of StyleGAN2 images and their reconstructions.

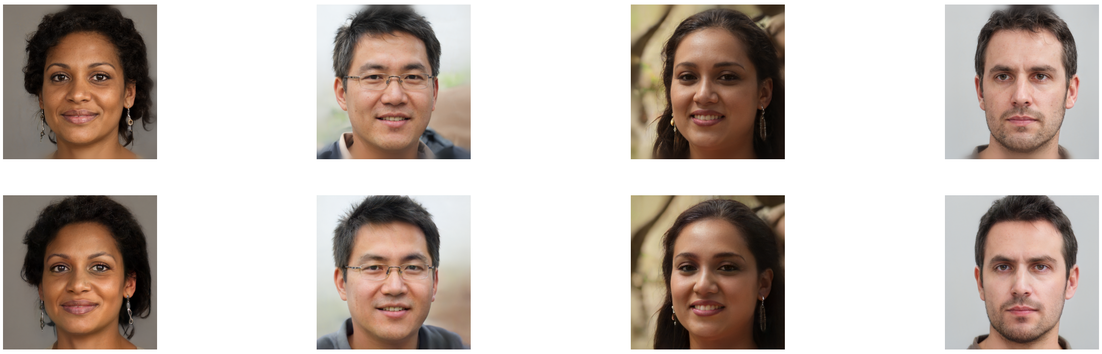


Figure 3. Examples of face images before (top row) and after (bottom row) the reconstruction using the inversion StyleGAN2 process available https://github.com/NVlabs/stylegan2-ada-pytorch/blob/main/projector.py (accessed on 30 December 2022).

Therefore, given a face image under analysis $\mathbf{x}_\mathrm{I}$, possibly coming from a variety of sources, we propose to compare it to its reconstructed version $\mathbf{x}_\mathrm{R}$. If $\mathbf{x}_\mathrm{I}$ has been actually synthesized by the considered generator G, its reconstruction $\mathbf{x}_\mathrm{R}$ should coincide or be very close to $\mathbf{x}_\mathrm{I}$, also depending on the effectiveness of the optimization strategy employed to solve the problem in (1). Conversely, when $\mathbf{x}_\mathrm{I}$ is not an output of the generator, the inversion process can only identify the closest reconstruction achievable through G.

3.2. Feature Extraction and Classification Process

After the inversion and reconstruction, we aim at jointly characterizing the visual appearance of $\mathbf{x}_\mathrm{I}$ and $\mathbf{x}_\mathrm{R}$, with the goal of predicting whether the former has been synthesized through G or not. Thus, in all cases, the objects of our analysis will be *images face pairs*, which we indicate as being *real* and *synthetic* when the related $\mathbf{x}_\mathrm{I}$ is real or synthetic, respectively. In particular, as depicted in Figure 4, we extract a feature representation separately from each image in the pair, thus obtaining $\mathbf{F}_\mathrm{I}$ and $\mathbf{F}_\mathrm{R}$. Then, a comparison operator between the two is applied, thus obtaining for each pair a single *differential feature vector*. Those are then used to train a classifier, to automatically characterize the differences resulting from the inversion process.

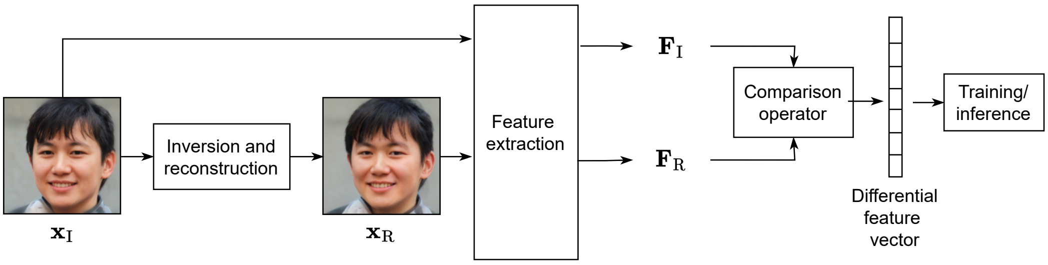
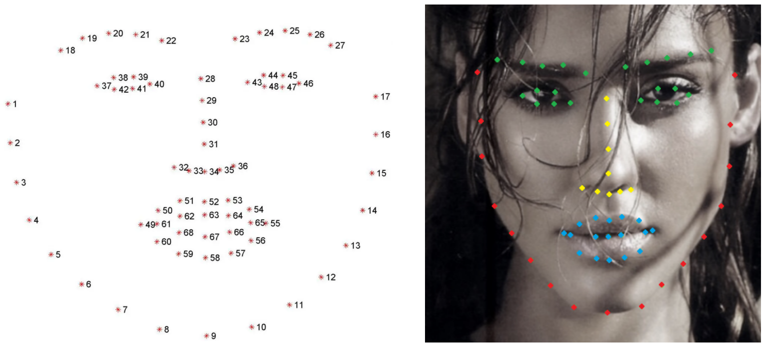


Figure 4. Pipeline of the comparison and training/classification processes.

We employ different feature extractors and comparison rules with the goal of capturing specific biometric traits, resulting in different types of differential feature vectors. In particular, we adopt as feature extractors both deep embeddings and handcrafted features proposed in the literature for automated face analysis tasks, namely:

- *FaceNet embeddings*: Proposed in [26], the FaceNet features are the best-performing ones on the LFW face recognition dataset [27] among the deep features selected in the Deepface toolbox (https://github.com/serengil/deepface (accessed on 30 December 2022)). In computing $\mathbf{F}_\mathrm{I}$ and $\mathbf{F}_\mathrm{R}$, we employ the original 512-dimensional FaceNet version and its compact 128-dimensional variant. In this case, the comparison is simply an element-wise difference in the module.
  We denote as $FN_{128} \in \mathbb{R}^{128}$ and $FN_{512} \in \mathbb{R}^{512}$ the two types of differential feature vectors obtained.
- *Facial landmarks*: proposed in [28] and available in the https://github.com/davisking/dlib (accessed on 30 December 2022) library, the landmark localization algorithm returns 68 facial landmarks related to key facial structures. Those can be further partitioned into different face areas (face line, eyebrows, eyes, nose, and mouth), as is shown in Figure 5. This feature extractor outputs the arrays $\mathbf{F}_\mathrm{I}$ and $\mathbf{F}_\mathrm{R}$ of size 68 × 2, containing row-wise, the 2D coordinates of the 68 landmarks, and we extract from them two types of differential feature vectors:
  - $LM_{68} \in \mathbb{R}^{68}$ contains the Euclidean distances between $\mathbf{F}_\mathrm{I}[i,:]$ and $\mathbf{F}_\mathrm{R}[i,:]$, $i = 1, \ldots 68$ (i.e., the 2D coordinates of corresponding landmarks in the two different faces);
  - $LM_{136} \in \mathbb{R}^{136}$ contains the differences in module between individual corresponding landmark coordinates $\mathbf{F}_\mathrm{I}[i,j]$ and $\mathbf{F}_\mathrm{R}[i,j]$, $i = 1, \ldots 68$, $j = 1, 2$.



(**a**) landmarks (**b**) landmarks detected

Figure 5. (**a**) Landmarks numbered from 1 to 68; (**b**) Landmarks detected on a face. We can identify 5 areas: face line (red landmarks [1–17]), eyebrows [18–27], and eyes [37–48], both in green (we call them eye area), nose [28–36] in yellow, and mouth [49–68] in blue.

4. Experimental Setup and Analysis of Results

We report the results of our experimental campaign on synthetic and real face images of different sources, and by employing different metrics and feature representations for the joint analysis of $\mathbf{x}_I$ and $\mathbf{x}_R$.

In particular, we considered the image data employed for the work [2], which have been made available by the authors (https://osf.io/ru36d/ (accessed on 30 December 2022)). They include synthetic images generated through StyleGAN2 (indicated as *SG2*), real images extracted from FFHQ (indicated as *FFHQ*), and the high-quality image dataset of human faces used for training StyleGAN2 (https://github.com/NVlabs/ffhq-dataset (accessed on 30 December 2022)).

Moreover, to diversify the data corpus and test generalization capabilities, we considered additional sets of real images coming from different sources, in particular:

- *CelebA*: a subset of https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html (accessed on 30 December 2022) [29], proposed for face detection and recognition, landmark localization, and face editing.
- *CelebHQ*: a subset of the https://mmlab.ie.cuhk.edu.hk/projects/CelebA/CelebAMask_HQ.html (accessed on 30 December 2022) dataset [30], proposed for evaluating algorithms in face parsing, recognition, and generation.
- *Caltech*: a subset of https://data.caltech.edu/records/6rjah-hdv18 (accessed on 30 December 2022) involving 27 subjects with different expressions and under different illumination conditions.
- *LFW*: a subset of the http://vis-www.cs.umass.edu/lfw/ (accessed on 30 December 2022) (LFW) dataset [27], a public benchmark for face verification.

Details about the data used in the experiments are reported in Figure 6.

	SYNTHETIC			REAL		
	SG2	*FFHQ*	*CelebA*	*CelebHQ*	*Caltech*	*LFW*
No. of images	400	400	390	397	446	398
Resolution	1024×1024	1024×1024	178×218	1024×1024	896×592	250×250
Format	PNG	PNG	PNG	PNG	JPEG	PNG

Figure 6. Summary of the face image data used in the experiments.

For each image, we applied the inversion process and obtained its reconstructed version; we used the default parameters of the inversion algorithms and fixed the random seed for reproducibility. As a pre-processing step for all images before the inversion, we detected the squared area containing the face through the https://github.com/davisking/dlib (accessed on 30 December 2022) library, and blurred the background outside that area to retain mostly face information in the input data. If needed, we resized the area (using the https://pillow.readthedocs.io/en/stable/ (accessed on 30 December 2022) library) to the resolution 1024 × 1024, which is the one accepted by the inversion algorithm. The average time for reconstructing an input image on an NVIDIA RTX 3090 GPU is 90 s.

Examples of input and reconstructed images are reported in Table 1. We also release the input and reconstructed images https://tinyurl.com/puusfcke (accessed on 30 December 2022).

Table 1. Examples of input face images and their reconstructions.

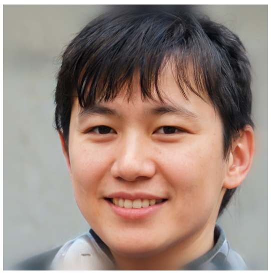
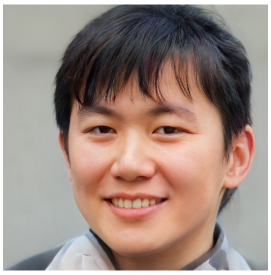
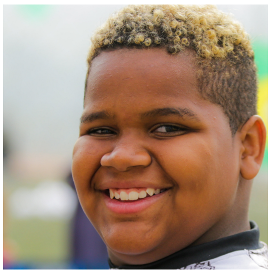
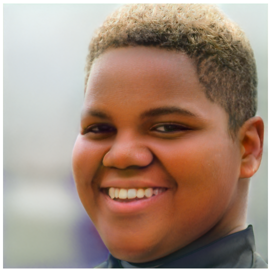
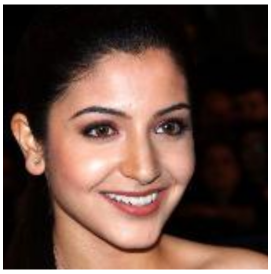
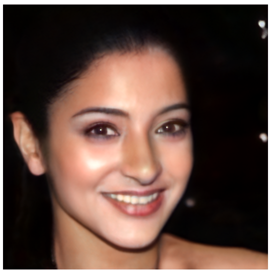
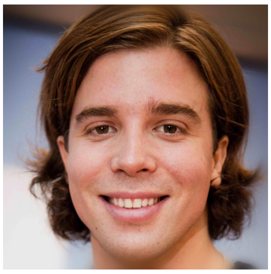
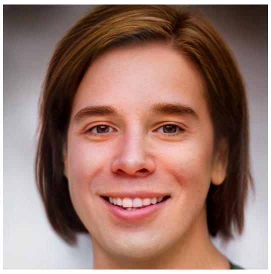
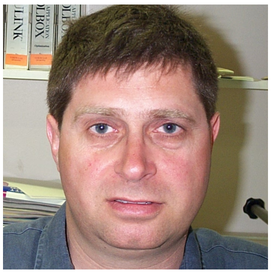
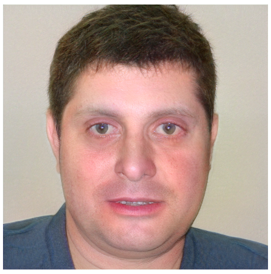
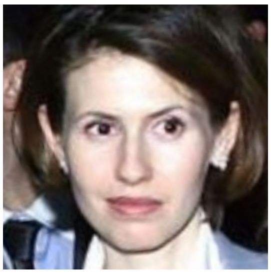
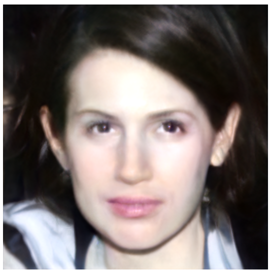
Dataset	$\mathbf{x_I}$	$\mathbf{x_R}$
SG2		
FFHQ		
CelebA		
CelebHQ		
Caltech		

Table 1. *Cont.*

Dataset	$\mathbf{x}_\mathrm{I}$	$\mathbf{x}_\mathrm{R}$
LFW		

From visual inspection, it can be noticed that the face attributes of the synthetic image are reconstructed very accurately, while more pronounced discrepancies in the biometric traits are present for real images. In the following, we jointly study the image faces given as input to the inversion process and their reconstructed counterparts, thus, $\mathbf{x}_\mathrm{I}$ and $\mathbf{x}_\mathrm{R}$.

4.1. Metrics-Based Analysis

First, we perform a similarity analysis between each image and its reconstructed counterpart. We considered the following metrics:

- Mean Squared Error (MSE): Computes the distance pixel-wise between the two images

$$MSE(\mathbf{x}_\mathrm{I}, \mathbf{x}_\mathrm{R}) = \frac{1}{MN} \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{3} (\mathbf{x}_\mathrm{I}(n,m,k) - \mathbf{x}_\mathrm{R}(n,m,k))^2$$

  where M,N are the dimensions of the image (equal for both images); $\mathbf{x}_\mathrm{I}$ is the RGB input image and its $\mathbf{x}_\mathrm{R}$ reconstructed version.
- Structure Similarity Index Method (SSIM): it is a perception-based model that takes into account the mean values and the variances of the two images

$$SSIM(\mathbf{x}_\mathrm{I}, \mathbf{x}_\mathrm{R}) = \frac{(2\mu_\mathrm{I}\mu_\mathrm{R} + c_1)(2\sigma_\mathrm{IR} + c_2)}{(\mu_\mathrm{I}^2 + \mu_\mathrm{R}^2 + c_1)(\sigma_\mathrm{I}^2 + \sigma_\mathrm{R}^2 + c_2)}$$

  with μ_I, μ_R being the mean values of the two images, σ_I, σ_R variance of the two images, and c_1 and c_2 being stabilization factors.
- Learned Perceptual Image Patch Similarity (LPIPS) (https://github.com/richzhang/PerceptualSimilarity (accessed on 30 December 2022)): it is proposed in [31] and used in [4] for the same purpose; it computes the similarity between the activations of two image patches for some pre-defined network. A low LPIPS score means that the image patches are perceptually similar.

The results are reported in Figure 7. We can notice that SSIM histograms do not show a clear distinction among different clusters. Indeed, SSIM is sensitive to the perceivable changes in terms of structural information, which are usually not noticeable in GAN-generated images. On the contrary, we observe that pairs deriving from real images yield generally higher MSEs than the ones derived from synthetic faces (red histogram), making it evident that reconstructing a pointwise equal image of an unknown target is much more difficult. The same happens for the LPIPS metrics, where, following what was observed in [4], the *SG2* images yield a higher similarity with their reconstructed counterparts.

Moreover, real images belonging to different datasets lead to different distributions, both in terms of LPIPS and MSE. In particular, it is interesting to observe that *FFHQ* images (blue histogram) present significantly lower values concerning other sources of real images: this may be related to the fact that those images were included in the training set of StyleGAN2, and thus, they are known to the generator.

4.2. Classification Results

We now report the results of the classification analysis performed according to the pipeline proposed in Figure 4. First, we plot the histogram of the MSE between $\mathbf{F}_\mathrm{I}$ and $\mathbf{F}_\mathrm{R}$ among the different datasets. In Figure 8, we observe that both FaceNet embeddings are able to improve the discrimination capability already observed on Figure 7. In this representation, the images generated with SG2 produce a clear peak around low MSE values, with a rather limited overlap with the other clusters. As expected, the *FFHQ* image pairs lie between synthetic samples and other real ones.

The charts reported in Figure 9 propose an analysis of the individual subsets of landmarks, according to the face area in which they belong (see Figure 5). These plots allow us to grasp the importance and the specific contribution of different sets of landmarks corresponding to different areas in the face. Indeed, the face line and eyes areas (eyes and eyebrows landmarks) clearly highlight the differences between real and synthetic pairs (see Figure 9b–d), while the nose and mouth areas are less effective in discrimination (Figure 9e–f). Anyway, the whole set of landmarks leads to the strongest separation, and is therefore used for further analysis.

The UMAP visualization of the FN_{512} and LM_{68} differential features (Figure 10) provides a 2D view of the distribution of different pairs: while pairs deriving from real images of different sources essentially overlap, real and synthetic pairs clearly tend to cluster together.

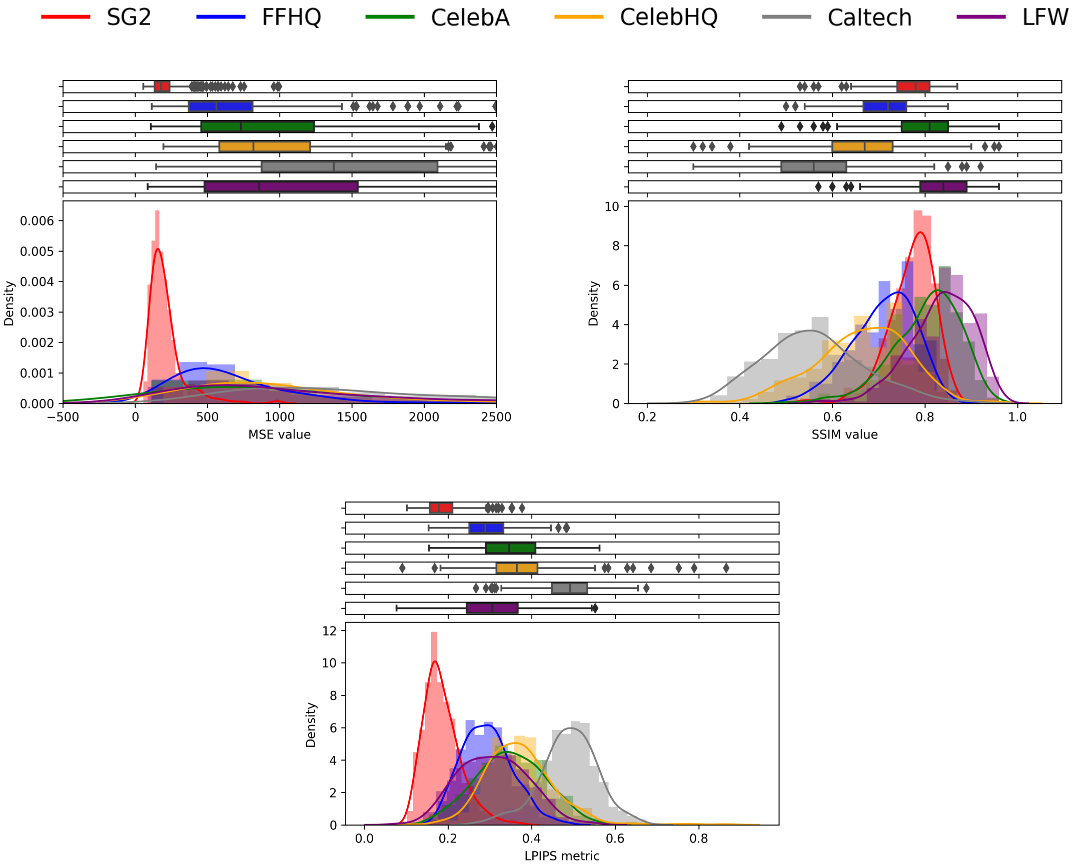
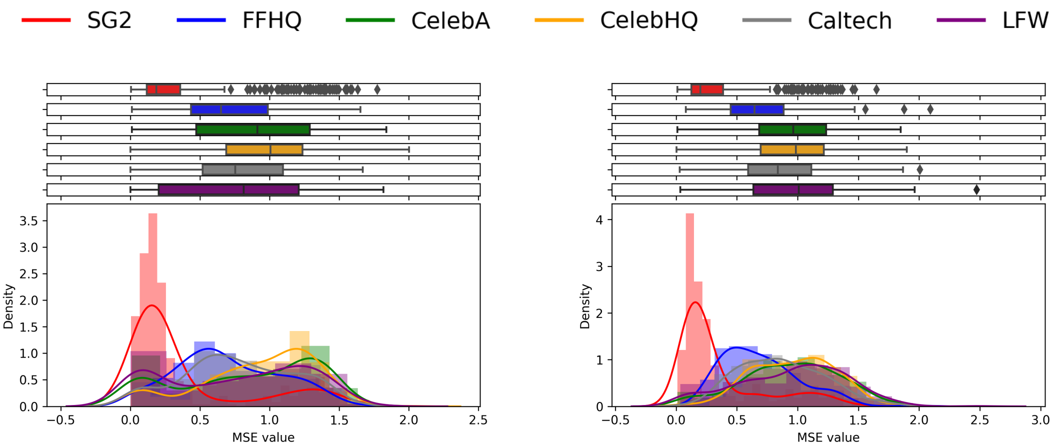
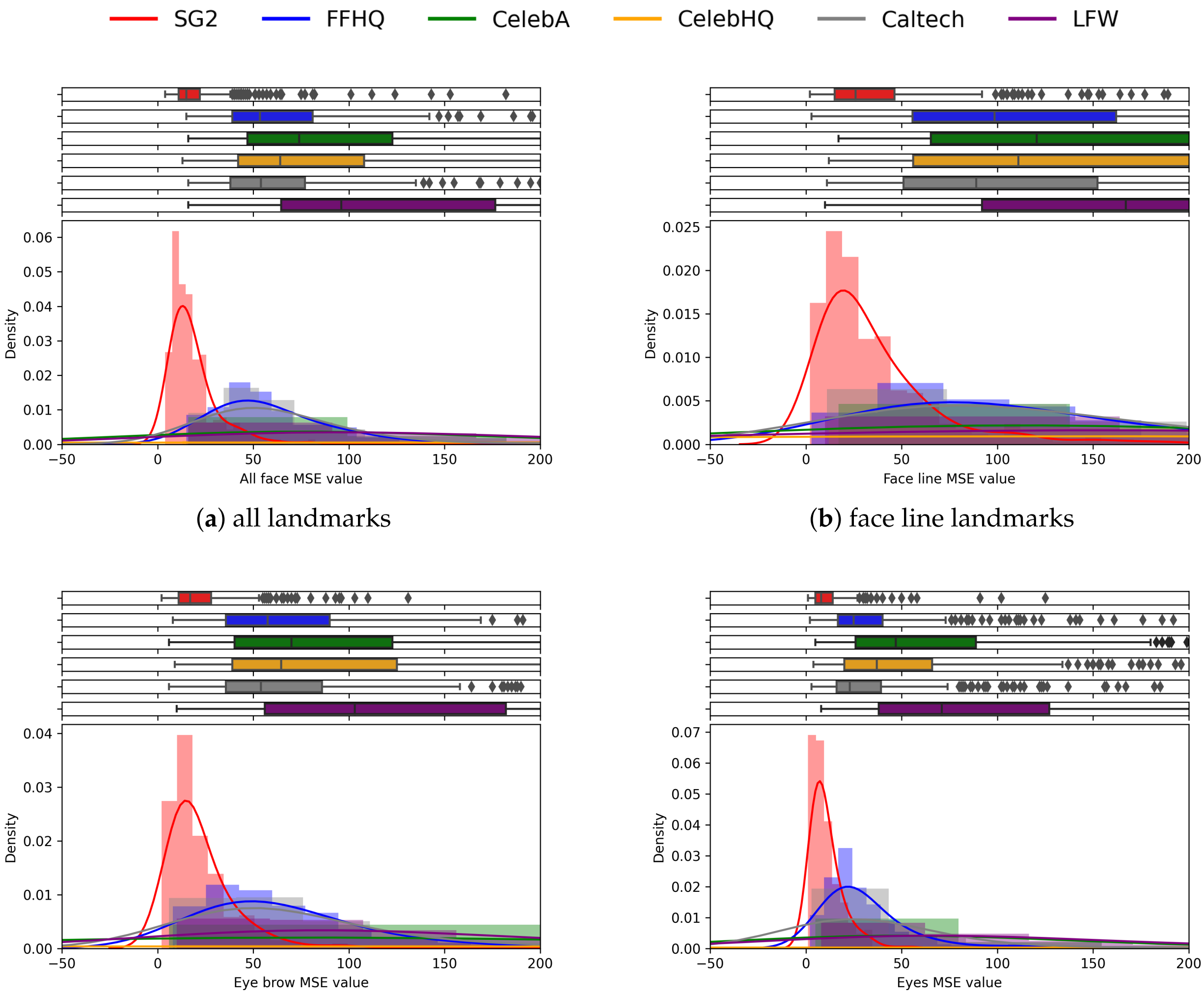


Figure 7. Histograms of different similarity metrics for images belonging to different datasets and their reconstructed counterparts. For each case, we report the density of the values of each similarity metric and their https://seaborn.pydata.org/generated/seaborn.boxplot.html (accessed on 30 December 2022) on top of the histogram, highlighting the interval between the first and third quartiles of each dataset (colored box), the median value of the distribution (vertical line within the colored box), and the outliers (grey diamonds).



(**a**) FaceNet128 (**b**) FaceNet512

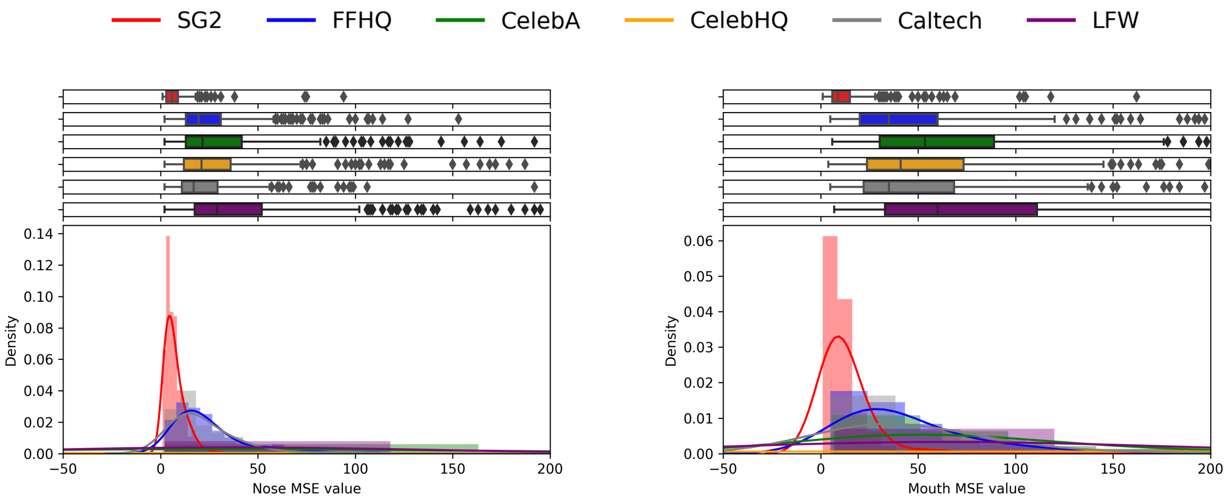
Figure 8. Comparison between FaceNet128 and FaceNet512 features using the MSE. FaceNet512 shows better results: the SG2 faces are well separated with respect to the other datasets, where the distributions tend to overlap each other. FaceNet128 shows an overlap between SG2 and LFW.



(**a**) all landmarks (**b**) face line landmarks

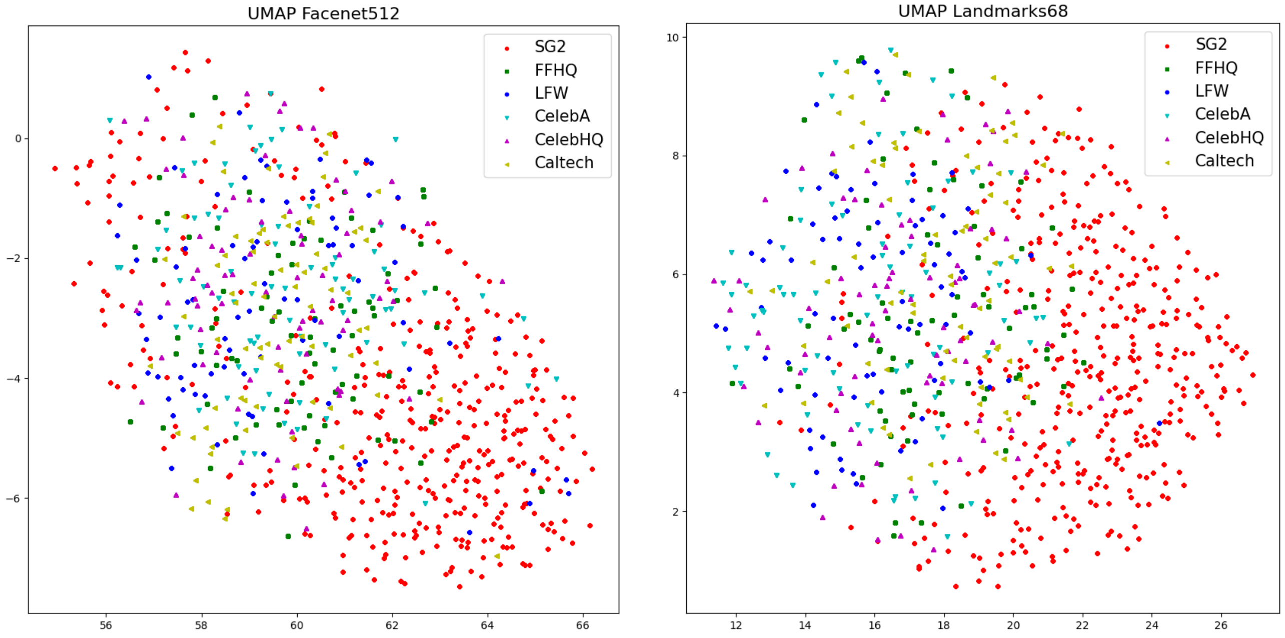
(**c**) eyebrows landmarks (**d**) eye landmarks

Figure 9. *Cont.*



(**e**) nose landmarks (**f**) mouth landmarks

Figure 9. Plots above visualize the histograms of the distributions of the landmarks of different areas of the face for the different datasets. The full set of landmarks shows better discrimination capability.



(**a**) UMAP for FN_{512} (**b**) UMAP for LM_{68}

Figure 10. Visualization of FN_{512} and LM_{68} differential feature vectors through the UMAP dimensionality reduction.

For performing training and inference, we split our datasets into fixed train and test sets. In particular, we used 80% of images from each dataset for training, and the remaining 20% for testing. Different classifiers are used for comparative analysis; in particular, Support Vector Machines (SVMs), Random Forest (RF), Logistic Regression (LR), and Multilayer Perceptrons (MLPs), as provided in https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (accessed on 30 December 2022) tool. We employed algorithms with default parameters and applied grid search for optimizing hyperparameters. In addition, we tested the Feedforward Neural Network (FNN) model provided by https://www.deeplearningwizard.com/deep_learning/practical_pytorch/pytorch_feedforward_neuralnetwork/ (accessed on 30 December 2022).

The results obtained with different types of differential vectors are reported in the following Table 2a–d. We first tested the datasets of real images individually against the *SG2* data (indicated as * vs. *SG2*), as well as their union (indicated as *All* vs. *SG2*), yielding six different settings reported row-wise in the tables.

We observe that the SVM provides, on average, better results, and also in front of limited computational complexity. In addition, we verified that the Radial Basis Function (RBF) kernel with hyperparameter $C = 1$ consistently yields the best performance; thus, we select it as the reference model for the following experimental analyses. In terms of computational efficiency, the training time of SVM models is in the order of milliseconds, thus, it is negligible with respect to the inversion time.

Interestingly, the landmark-based differential analysis yields substantially higher discrimination capabilities with respect to the FaceNet-based one, despite their generally lower dimensionality. In particular, they perform exceptionally well on the *LFW* pairs, which are the more critical case for the FaceNet representations. For a better understanding, we report in Figure 11 a comparative visualization of the landmarks detected in the input and reconstructed faces, and we plot them together to visualize the misalignment. It can be seen that the synthetic StyleGAN2 image pairs present almost overlapping landmarks, while the real ones show irregular displacements of individual landmarks.

Table 2. Accuracy obtained with different data and classifiers (reported in percentage).

	SVM	**RF**	**LR**	**MLP**	**FNN**
FFHQ vs. *SG2*	80.63	76.88	70.63	70.63	79.38
CelebA vs. *SG2*	82.28	82.91	69.62	69.62	82.91
CelebHQ vs. *SG2*	88.05	86.16	77.99	77.99	86.79
LFW vs. *SG2*	76.88	74.38	71.25	73.13	75.63
Caltech vs. *SG2*	89.38	81.88	81.25	81.25	83.13
All vs. *SG2*	78.13	75.00	71.88	73.13	77.50

(a) FN_{128}

	SVM	**RF**	**LR**	**MLP**	**FNN**
FFHQ vs. *SG2*	81.88	78.75	73.75	73.13	80.00
CelebA vs. *SG2*	88.61	84.81	83.54	81.01	88.61
CelebHQ vs. *SG2*	86.79	84.91	81.13	79.25	86.16
LFW vs. *SG2*	79.38	78.13	77.50	72.50	81.25
Caltech vs. *SG2*	85.63	83.75	85.00	86.25	86.25
All vs. *SG2*	78.13	78.75	76.88	76.25	78.75

(b) FN_{512}

	SVM	**RF**	**LR**	**MLP**	**FNN**
FFHQ vs. *SG2*	87.50	84.37	83.12	86.87	82.50
CelebA vs. *SG2*	89.24	91.14	88.60	87.34	87.34
CelebHQ vs. *SG2*	89.30	88.67	86.79	83.01	83.02
LFW vs. *SG2*	95.59	95.59	89.93	94.96	94.33
Caltech vs. *SG2*	87.50	87.50	86.25	87.50	87.50
All vs. *SG2*	89.37	88.12	85.00	85.00	84.37

(c) LM_{68}

	SVM	**RF**	**LR**	**MLP**	**FNN**
FFHQ vs. *SG2*	88.75	84.37	80.62	85.00	85.62
CelebA vs. *SG2*	89.87	92.40	87.34	84.17	89.87
CelebHQ vs. *SG2*	89.30	88.67	81.76	84.90	84.90
LFW vs. *SG2*	94.96	93.71	84.27	90.56	94.96
Caltech vs. *SG2*	90.62	90.00	86.87	90.00	87.50
All vs. *SG2*	88.75	87.50	82.50	83.12	85.00

(d) LM_{136}

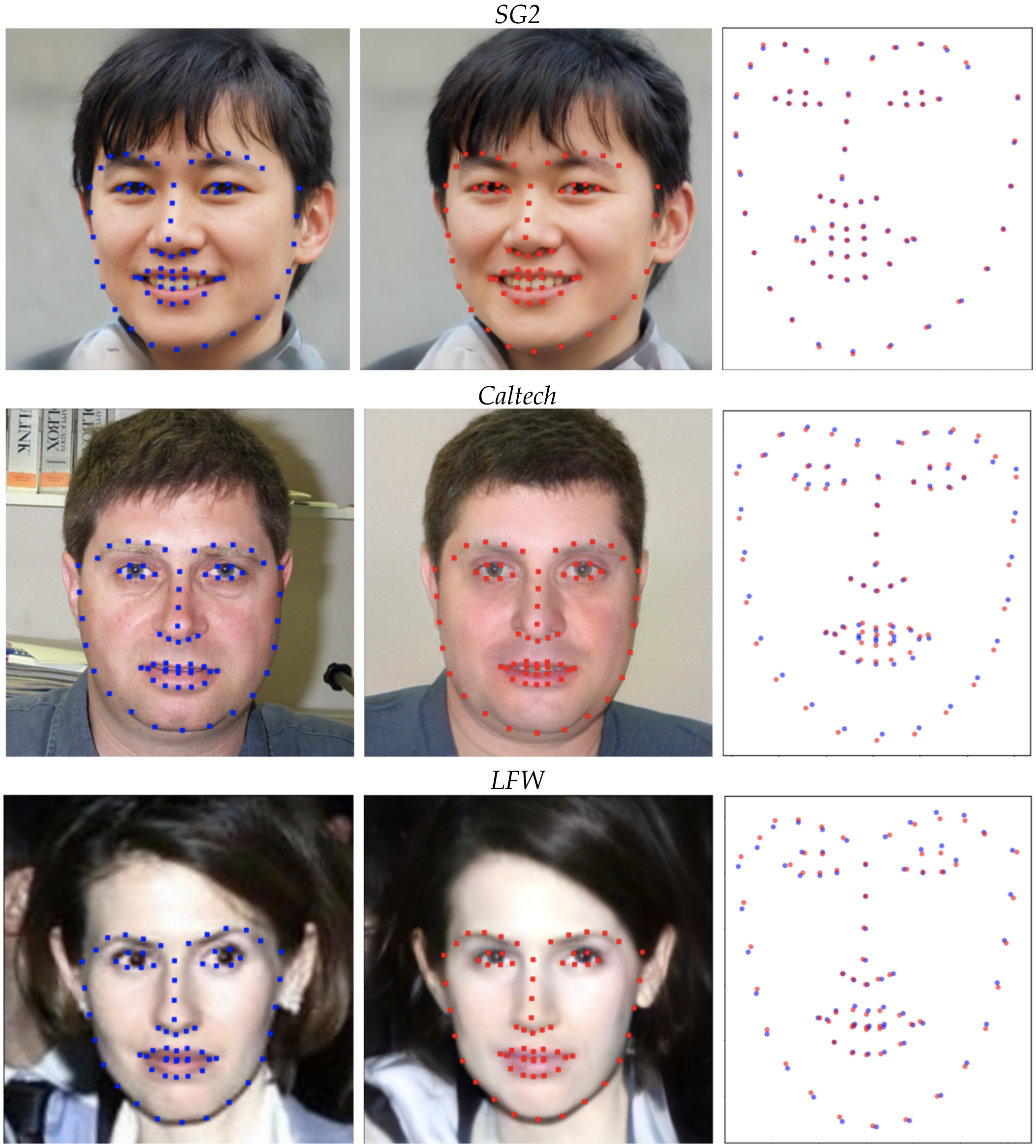
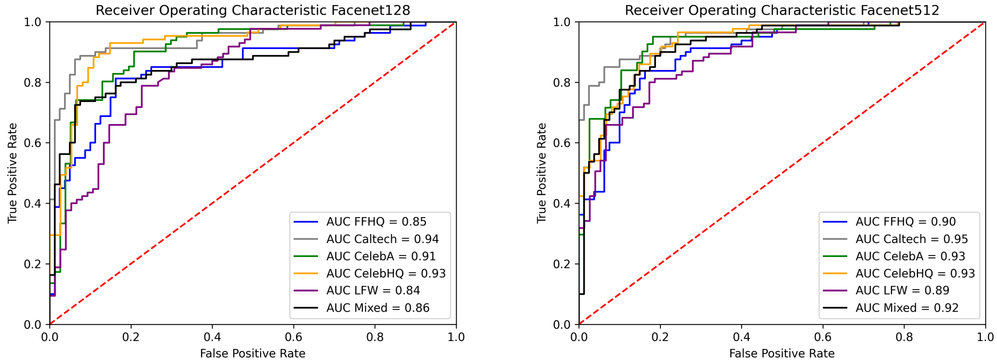


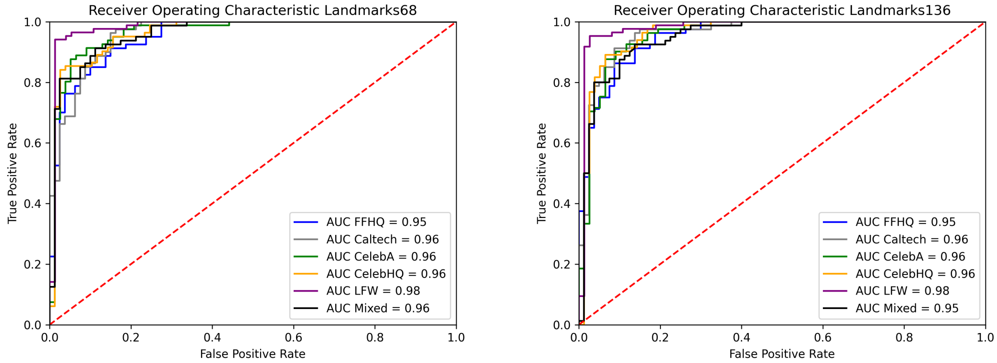
Figure 11. Visualization of the landmarks detected on the input and the reconstructed images from different datasets. In each case, the left image is the input one with the detected landmarks marked in blue, and the central one is the corresponding reconstruction with the detected landmarks marked in red. On the right, the two sets of landmarks are reported on the same spatial grid, so that their displacement can be visualized.

In general, the *FFHQ* pairs seem to be the harder ones to distinguish from *SG2* pairs also for landmark-based features, possibly because the former were employed for training the StyleGAN2 generator. This is also observed in Figures 12 and 13, where the ROC curves of the different classification scenarios and feature representations are reported. Even if the results are very good in all cases, Table 3 shows that AUC values for the LM_{68} and LM_{136} case remain lower for the *FFHQ* data with respect to other real images.



(**a**) ROC for FN_{128}. (**b**) ROC for FN_{512}.

Figure 12. Comparison of the ROC curves obtained with the SVM model using Facenet features.



(**a**) ROC for LM_{68}. (**b**) ROC for LM_{136}.

Figure 13. Comparison of the ROC curves obtained with the SVM model using landmark-based features.

Table 3. AUC values for the SVM classifiers.

	$\boldsymbol{FN_{128}}$	$\boldsymbol{FN_{512}}$	$\boldsymbol{LM_{68}}$	$\boldsymbol{LM_{136}}$
FFHQ vs. *SG2*	0.85	0.90	0.95	0.95
CelebA vs. *SG2*	0.91	0.93	0.96	0.96
CelebHQ vs. *SG2*	0.93	0.93	0.96	0.96
LFW vs. *SG2*	0.84	0.89	0.98	0.98
Caltech vs. *SG2*	0.94	0.95	0.96	0.96
All vs. *SG2*	0.86	0.92	0.96	0.95

4.3. Robustness Analysis

An aspect of high practical relevance is whether synthetic images are still identified through the inversion-based analysis, even though they are not the direct output of the generator, but undergo successive post-processing. An advantage of facial landmarks is the fact that their detection and localization are rather robust to the operations applied to the images under analysis. FaceNet embeddings are also designed to generalize to different face image scales and conditions.

Since handling the variety of (even slight) potential operations is a known issue for data-driven techniques based on learned features, we now assess the robustness of the classifiers developed in Section 4.2 when training and testing data are not aligned in terms of post-processing. In this view, we study three routinely applied operations in the lifecycle of digital images, namely resizing, JPEG compression, and social network sharing. For the sake of conciseness, we focus on the case of *FFHQ* vs. *SG2*, which is the most critical one for the best-performing features.

4.3.1. Resizing

We have tested our discrimination models with inputs at different resolution levels by downscaling and upscaling the images. In particular, we rescale the entire datasets at different scaling factors $\{0.3, 0.7, 0.8, 0.9, 1.1, 1.2, 1.3\}$, we apply the inversion/reconstruction process for each case, and finally, we compute the differential feature vectors, FN_{128}, FN_{512}, LM_{68}, and LM_{136}. The library used for the resize process is https://pillow.readthedocs.io/en/stable/ (accessed on 30 December 2022), with the nearest neighbor resample parameter set (`PIL.Image.NEAREST` in the code). Since the StyleGAN2 inversion algorithm requires inputs with fixed size 1024 × 1024 pixels, when inverting images of different resolutions, a further cropping/rescaling operation is needed to meet this requirement. In doing so, some details of the image change in terms of quality (see Figure 14), making the discrimination between real images and fake images in principle harder.

After having scaled all the images, we trained and tested the models with these resized examples. Tables 4–7 report the results obtained, where the training scaling factors are reported row-wise, and the testing scaling factors column-wise. Rows and columns with a scaling factor of 1.0 correspond to the baseline case (no scaling), where no post-processing is applied to either training or testing images. We notice that the accuracies are generally preserved and they present no dramatic drops, but rather, oscillations around the aligned cases corresponding to the diagonal values. For the majority of classifiers, the average variation over different testing sets does not exceed 2%.

Among the different representations, the FaceNet-based features seem to struggle more with the upscaled images rather than the downscaled ones, occasionally decreasing below 80%. This behavior is reversed for landmark-based features, for which the performances are more stable for upscaling factors and more sensible and oscillatory for downscaling ones. In both cases, the dimensionality of the differential vectors does not have a significant impact.

Table 4. Accuracy obtained by FN_{128} with different resizing factors (reported in percentage).

Train/Test	0.3	0.7	0.8	0.9	1.0	1.1	1.2	1.3
0.3	84.35	82.50	83.75	84.37	84.38	81.87	81.88	81.25
0.7	85.00	85.00	84.37	88.13	85.00	82.50	83.12	79.36
0.8	83.75	85.00	85.00	86.25	81.88	82.50	81.25	81.25
0.9	83.12	80.62	81.88	86.25	80.00	78.75	78.75	78.13
1.0	83.12	83.75	80.00	82.50	80.63	80.00	78.13	77.50
1.1	83.12	82.50	82.50	83.75	80.62	81.25	78.75	80.00
1.2	84.61	85.25	83.33	87.18	84.62	80.77	83.97	82.69
1.3	82.70	80.77	80.12	85.26	82.05	83.97	83.33	80.77

Table 5. Accuracy obtained by FN_{512} with different resizing factors (reported in percentage).

Train/Test	0.3	0.7	0.8	0.9	1.0	1.1	1.2	1.3
0.3	86.88	83.13	78.75	82.50	84.38	81.88	80.00	83.75
0.7	85.00	80.62	78.12	81.25	81.88	80.63	80.63	80.63
0.8	85.00	78.12	76.25	80.63	81.25	78.75	78.75	80.00
0.9	83.75	81.25	80.00	80.00	82.50	80.00	78.75	80.63
1.0	85.63	81.25	81.25	78.13	81.88	80.00	80.00	77.50
1.1	83.75	80.63	80.00	82.50	83.13	83.13	80.00	80.63
1.2	81.41	78.21	76.92	78.85	80.79	77.56	80.00	80.63
1.3	80.13	75.64	81.41	78.85	83.33	80.77	78.85	80.75

Table 6. Accuracy obtained by LM_{68} with different resizing factors (reported in percentage).

Train/Test	0.3	0.7	0.8	0.9	1.0	1.1	1.2	1.3
0.3	88.13	86.87	83.75	85.62	81.76	85.00	83.33	85.25
0.7	86.87	86.25	84.37	90.62	85.62	85.00	91.66	90.38
0.8	90.65	88.12	86.25	91.25	87.50	86.25	91.02	87.82
0.9	90.00	87.50	87.50	90.62	86.25	86.25	89.10	90.38
1.0	90.00	88.12	86.25	90.62	87.50	85.32	90.25	88.21
1.1	88.75	88.75	85.62	88.75	85.00	87.50	89.10	87.17
1.2	90.00	90.62	89.37	91.25	88.12	90.00	92.94	91.66
1.3	90.00	88.75	86.88	90.65	86.88	90.00	92.31	90.38

Table 7. Accuracy obtained by LM_{136} with different resizing factors (reported in percentage).

Train/Test	0.3	0.7	0.8	0.9	1.0	1.1	1.2	1.3
0.3	85.62	86.87	83.75	86.87	86.25	85.00	87.18	86.53
0.7	88.75	89.37	85.00	90.62	88.12	88.12	91.02	91.02
0.8	91.25	90.62	83.75	92.50	90.00	90.62	90.38	89.10
0.9	91.25	87.75	85.62	90.00	86.87	88.75	89.10	89.74
1.0	89.37	86.87	85.00	90.00	88.75	89.37	90.38	89.10
1.1	90.62	88.75	85.00	89.37	87.50	88.12	90.38	88.46
1.2	92.50	91.25	86.25	90.62	88.75	90.62	91.66	89.10
1.3	93.75	90.62	88.12	91.25	88.12	93.75	94.87	92.30

4.3.2. JPEG Compression

We apply the same robustness analysis for the JPEG compression at different quality factors $\{100, 95, 90, 80, 70\}$. Examples of compressed images and their reconstructions are reported in Figure 15.

The results are reported in Tables 8–11. As for the resizing section, the library used for the compression process is https://pillow.readthedocs.io/en/stable/ (accessed on 30 December 2022). We varied the quality parameter of the saved image. The baseline case is reported in the 'NO COMP' rows and columns. Additionally, in this case, all of the feature representations generally retain their accuracies when the training and testing sets are misaligned, as most of the models have an average deviation from the aligned case below 2%.

Interestingly, when observing the results column-wise, we notice that for FaceNet-based features, a stronger JPEG compression consistently degrades the average performance of the classifiers; as opposed to that, LM_{68} and LM_{136} fully retain their accuracy, thus strengthening the observation that such semantic cues yield an improved robustness to post-processing.

Table 8. Accuracy obtained by FN_{128} with different JPEG quality factors (reported in percentage).

Train/Test	NO COMP	100	95	90	80	70
NO COMP	80.62	80.62	78.12	76.25	79.37	76.87
100	78.12	78.12	72.50	73.75	74.37	75.00
95	78.75	80.62	76.87	73.12	79.37	76.87
90	82.50	83.12	80.62	79.37	84.37	78.75
80	81.25	76.87	76.25	74.37	81.25	77.50
70	78.12	80.62	75.62	76.87	78.10	74.37

Table 9. Accuracy obtained by FN_{512} with different JPEG quality factors (reported in percentage).

Train/Test	NO COMP	100	95	90	80	70
NO COMP	81.87	80.62	78.75	78.12	80.00	81.25
100	79.37	77.50	76.87	75.62	76.87	77.50
95	78.12	79.37	77.50	76.87	76.25	76.87
90	81.87	82.50	81.25	78.75	80.62	81.87
80	83.12	82.50	76.87	77.50	78.75	81.87
70	85.00	85.62	83.75	83.12	84.37	83.12

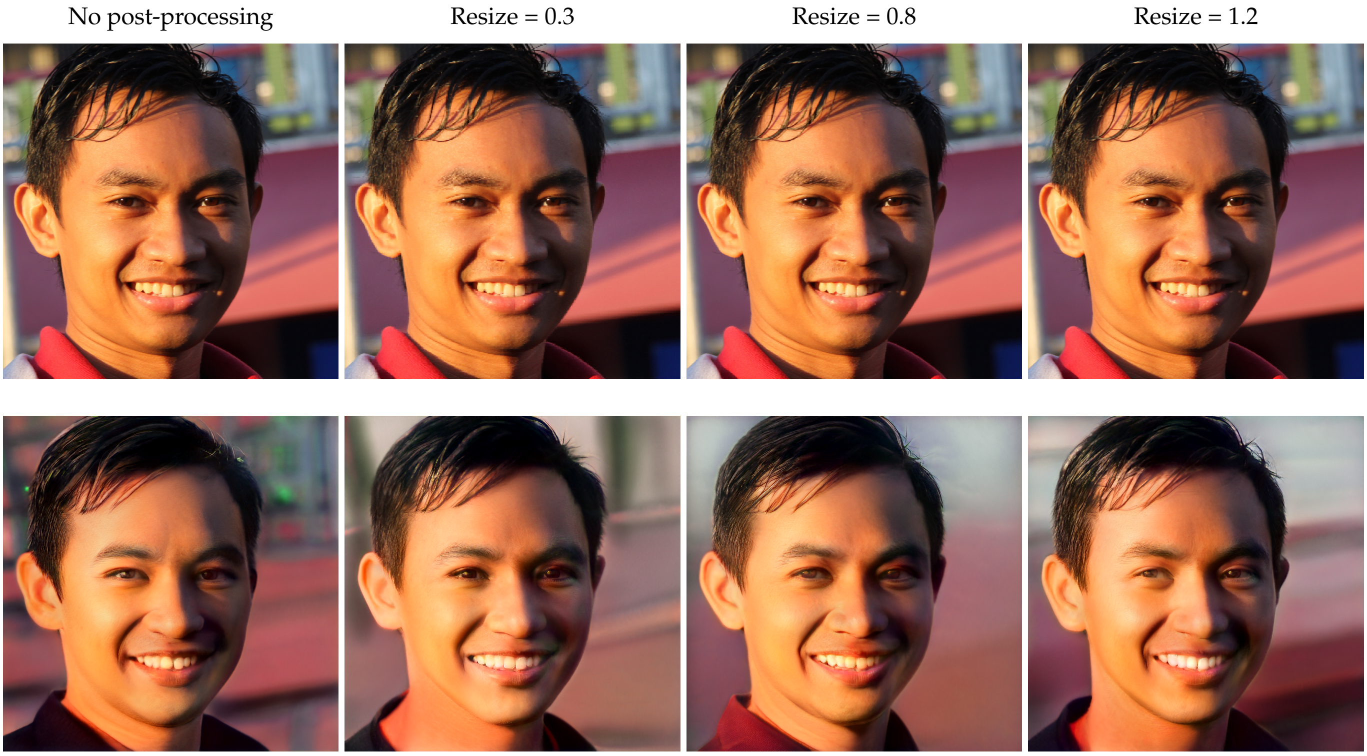


Figure 14. Examples of face images before (top row) and after (bottom row) the reconstruction when different resizing factors are applied as post-processing.

Table 10. Accuracy obtained by LM_{68} with different JPEG quality factors (reported in percentage).

Train/Test	**NO COMP**	**100**	**95**	**90**	**80**	**70**
NO COMP	87.50	84.37	90.00	85.62	89.37	90.00
100	85.62	88.12	86.87	86.25	88.12	88.12
95	85.00	85.00	86.87	85.62	87.50	86.25
90	83.75	85.62	85.62	82.50	86.87	86.25
80	87.50	90.00	88.75	85.62	89.37	90.00
70	85.00	89.37	86.25	89.37	90.00	90.62

Table 11. Accuracy obtained by LM_{136} with different JPEG quality factors (reported in percentage).

Train/Test	**NO COMP**	**100**	**95**	**90**	**80**	**70**
NO COMP	88.75	88.12	88.75	85.00	90.62	88.75
100	84.37	88.75	88.75	86.87	87.50	88.75
95	86.25	86.87	90.62	88.75	88.75	86.87
90	86.87	88.12	90.62	86.87	89.37	88.12
80	87.50	91.25	90.62	86.87	89.37	91.25
70	87.50	90.62	90.00	89.37	90.00	90.62

4.3.3. Social Network Sharing

The identification of synthetic media over social networks is a well-known challenge in media forensics [32], and an open issue for GAN-generated image detection [33]. Social media typically apply custom data compression algorithms to reduce the size and the quality of the images to be stored on data centers or costumer's devices, thus hindering post hoc analyses.

We then test the capabilities of the developed classifiers to generalize to the image data shared on social networks. It is worth noticing that in this case, the models are exactly the ones considered in the classifiers developed in Section 4.2; thus, they are trained entirely on images with no sharing operations. Instead, the testing set is a subset of the recently published https://zenodo.org/record/7065064#.Y2to3ZzMKdZ (accessed on 30

December 2022) dataset, which is composed of StyleGAN2 images and real images (extracted from FFHQ) before and after the upload and download from three different social networks: Facebook, Telegram, and Twitter. We randomly select 100 synthetic and 100 real images and extract their shared versions through the three platforms.

Examples of shared images and their reconstructions are reported in Figure 16.

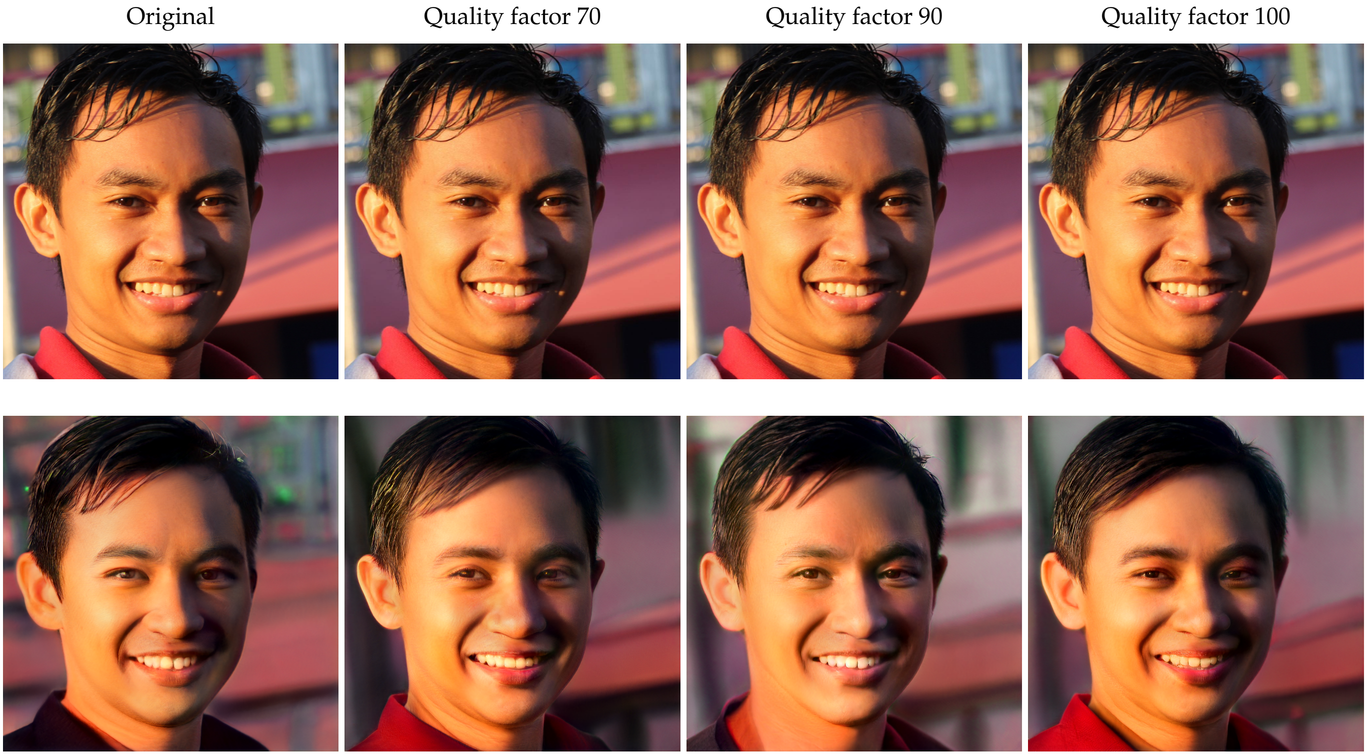


Figure 15. Examples of face images before (top row) and after (bottom row) the reconstruction when different JPEG quality factors are applied as post-processing.

After inversion, we obtain a reconstruction for each of them. If needed, a rescaling operation is applied to fit the input size of the inversion algorithm. We then extract the feature representations and differential vectors, and test them through the corresponding classifiers already trained in Section 4.2 for the *FFHQ* vs. *SG2* scenario.

The results are reported in Table 12, where the accuracies of each binary classification scenario (one for each platform) are reported column-wise. As highlighted in [33], all platforms apply JPEG compression (quality factor between 80 and 90), and Facebook also resizes the images by a 0.7 factor.

Interestingly, the landmark-based features yield remarkable performance in all cases, thus demonstrating a high robustness against this realistic kind of post-processing. In particular, they achieve a maximum accuracy in the Facebook case, which is the more critical one for FaceNet-based features, and also for the general purpose deep networks analyzed in [33].

Table 12. Accuracy of the different classifiers (SVM model) on a subset of the TrueFace dataset, including images shared through different social media platforms (reported in percentage).

	Facebook	**Telegram**	**Twitter**	**All**
FN_{128}	73	85	77	80
FN_{512}	74	82	78	76
LM_{68}	96	97	98	96
LM_{136}	100	98	96	97

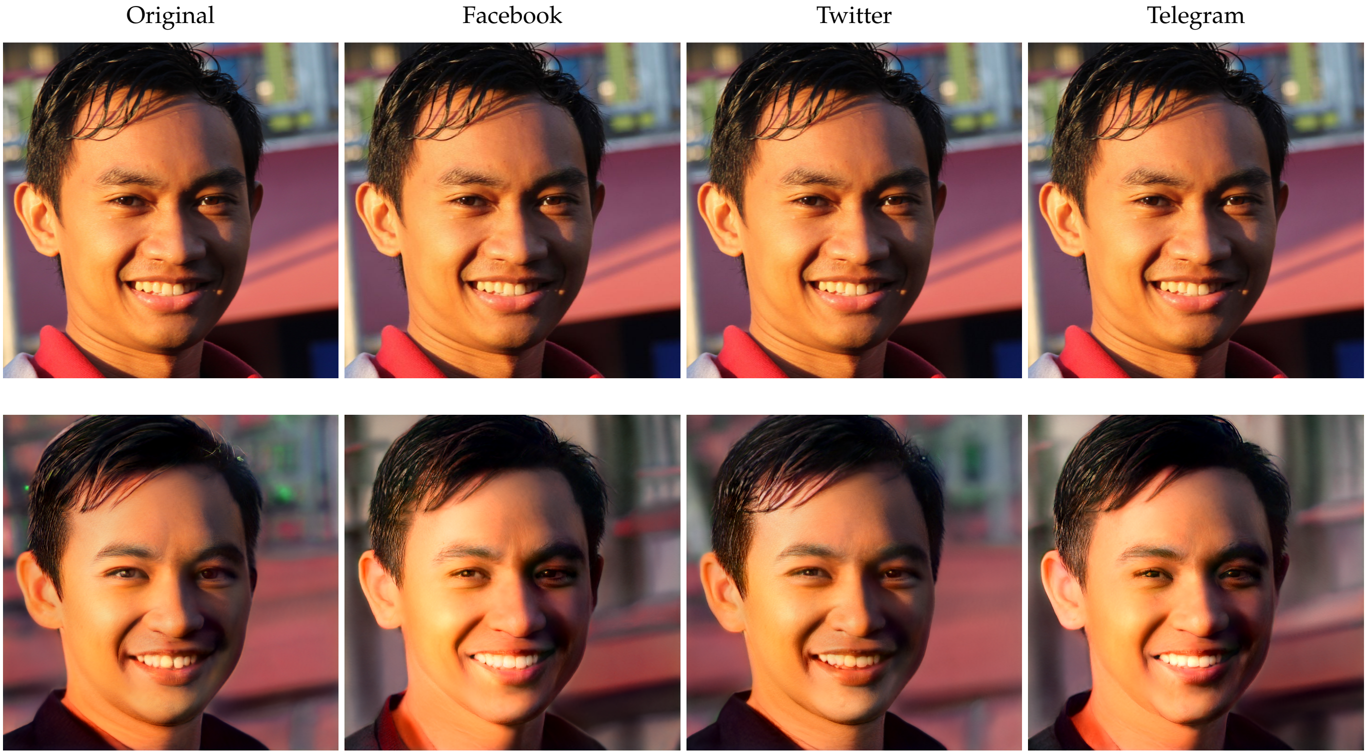


Figure 16. Shared images before (top row) and after (bottom row) the reconstruction.

5. Conclusions

We have explored a forensic detection strategy for identifying synthetic face images based on the inversion of the GAN synthesis process. The experimental results demonstrate that a proper biometric comparison between the image under investigation and its reconstruction through an inversion algorithm allows for distinguishing images that have been synthesized by the considered generator and those who are not. In particular, our analysis shows that landmark-based feature representations are particularly effective for this purpose.

A desirable aspect of such an approach is that it is not purely inductive, but is based on the very architecture of the generation methods. Moreover, the best-performing features refer to an explicit face model, and they express a biometric reconstruction dissimilarity that can be better interpreted with respect to deep representations.

On the other hand, a limitation of this approach is that it assumes prior knowledge of the candidate generator, for which an inversion procedure needs to be devised. In particular, this work focuses on a powerful yet single generator. Extensions of this work would consider more general scenarios where the inversion-based comparison is tested against multiple latest generators, such as StyleGAN3 [34] and EG3D [35]. This would also include dealing with more comprehensive data corpora with diverse facial attributes in terms of expression, gender or age, as well as more generative models that are trained to synthesize other objects beyond faces. Moreover, an effective fusion of our approach with data-driven techniques would be a promising direction for future investigations. In addition, an open research question is to which extent inversion-based techniques can be applied to generative models operating in domains other then the visual one, such as speech [36], text [37], or raw tabular data [38]. In this respect, we expect that the main concept of the paper still remains valid, while the metrics should be suitably revised to capture the most significant domain-sensitive differences.

Author Contributions: Conceptualization, C.P.; Data curation, F.L., D.L., and G.A.; Methodology, C.P., F.L., D.L., G.A., G.B., and F.D.N.; Project administration, C.P.; Software, F.L., D.L., and G.A.; Supervision, C.P., G.B., and F.D.N.; Writing—original draft, C.P., F.L., D.L., and G.A.; Writing—review and editing, G.B. and F.D.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding by the project PREMIER (PREserving Media trustworthiness in the artificial Intelligence ERa) under contract PRIN 2017 2017Z595XS-001, funded by the Italian Ministry of University and Research, from the TruBlo consortium as part of the TrueBees project (https://www.trublo.eu/truebees/ (accessed on 30 December 2022)) funded by TruBlo under the Europe's Horizon 2020 programme (grant agreement No. 957228), and by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112090136.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset of reconstructed images is available at https://tinyurl.com/puusfcke (accessed on 30 December 2022).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lago, F.; Pasquini, C.; Böhme, R.; Dumont, H.; Goffaux, V.; Boato, G. More Real Than Real: A Study on Human Visual Perception of Synthetic Faces. *IEEE Signal Process. Mag.* **2021**, *39*, 109–116. [CrossRef]
2. Nightingale, S.J.; Farid, H. AI-synthesized faces are indistinguishable from real faces and more trustworthy. *Proc. Natl. Acad. Sci. USA* **2022**, *119*, e2120481119. [CrossRef] [PubMed]
3. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.
4. Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; Aila, T. Analyzing and Improving the Image Quality of StyleGAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
5. Dang-Nguyen, D.T.; Boato, G.; De Natale, F.G. 3D-model-based video analysis for computer generated faces identification. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1752–1763. [CrossRef]
6. Bonomi, M.; Pasquini, C.; Boato, G. Dynamic texture analysis for detecting fake faces in video sequences. *J. Vis. Commun. Image Represent.* **2021**, *79*, 103239. [CrossRef]
7. Dang-Nguyen, D.; Boato, G.; De Natale, F. Identify computer generated characters by analysing facial expressions variation. In Proceedings of the IEEE International Workshop on Information Forensics and Security, Tenerife, Spain, 2–5 December 2012; pp. 252–257.
8. Gragnaniello, D.; Cozzolino, D.; Marra, F.; Poggi, G.; Verdoliva, L. Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. In Proceedings of the IEEE International Conference on Multimedia and Expo, Shenzhen, China, 5–9 July 2021; pp. 1–6.
9. Marra, F.; Saltori, C.; Boato, G.; Verdoliva, L. Incremental learning for the detection and classification of GAN-generated images. In Proceedings of the IEEE International Workshop on Information Forensics and Security, Delft, The Netherlands, 9–12 December 2019; pp. 1–6.
10. Marra, F.; Gragnaniello, D.; Verdoliva, L.; Poggi, G. Do GANs leave artificial fingerprints? In Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval, San Jose, CA, USA, 28–30 March 2019; pp. 506–511.
11. Wang, S.Y.; Wang, O.; Zhang, R.; Owens, A.; Efros, A.A. CNN-generated images are surprisingly easy to spot... for now. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 8695–8704.
12. Xia, W.; Zhang, Y.; Yang, Y.; Xue, J.H.; Zhou, B.; Yang, M.H. GAN Inversion: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 1–17. [CrossRef]
13. Nataraj, L.; Mohammed, T.M.; Manjunath, B.S.; Chandrasekaran, S.; FlennerJawadul, A.; Bappy, H.; Roy-Chowdhury, A.K. Detecting GAN generated Fake Images using Co-occurrence Matrices. *Electron. Imaging* **2019**, *2019*, 532-1. [CrossRef]
14. Wang, R.; Juefei-Xu, F.; Ma, L.; Xie, X.; Huang, Y.; Wang, J.; Liu, Y. FakeSpotter: A Simple yet Robust Baseline for Spotting AI-Synthesized Fake Faces. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Yokohama, Japan, 7–15 January 2020.
15. Marcon, F.; Pasquini, C.; Boato, G. Detection of Manipulated Face Videos over Social Networks: A Large-Scale Study. *J. Imaging* **2021**, *7*, 193. [CrossRef] [PubMed]
16. Dong, X.; Miao, Z.; Ma, L.; Shen, J.; Jin, Z.; Guo, Z.; Teoh, A.B.J. Reconstruct face from features based on genetic algorithm using GAN generator as a distribution constraint. *Comput. Secur.* **2023**, *125*, 103026. [CrossRef]

17. Albright, M.; McCloskey, S. Source Generator Attribution via Inversion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–20 June 2019.
18. Scherhag, U.; Rathgeb, C.; Merkle, J.; Busch, C. Deep Face Representations for Differential Morphing Attack Detection. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3625–3639. [CrossRef]
19. Autherith, S.; Pasquini, C. Detecting morphing attacks through face geometry. *J. Imaging* **2020**, *6*, 115. [CrossRef] [PubMed]
20. Chen, B.; Tang, G.; Sun, L.; Mao, X.; Guo, S.; Zhang, H.; Wang, X. Detection of GAN-Synthesized Image Based on Discrete Wavelet Transform. *Secur. Commun. Netw.* **2021**, *2021*, 5511435.
21. Wang, J.; Tondi, B.; Barni, M. An Eyes-Based Siamese Neural Network for the Detection of GAN-Generated Face Images. *Front. Signal Process.* **2022**, 45. [CrossRef]
22. Agarwal, S.; Farid, H. Detecting deep-fake videos from aural and oral dynamics. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 981–989.
23. Schwarcz, S.; Chellappa, R. Finding facial forgery artifacts with parts-based detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 933–942.
24. Ju, Y.; Jia, S.; Ke, L.; Xue, H.; Nagano, K.; Lyu, S. Fusing Global and Local Features for Generalized AI-Synthesized Image Detection. *arXiv* **2022**, arXiv:2203.13964R.
25. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In *Proceedings of the Advances in Neural Information Processing Systems*; Ghahramani, Z.; Welling, M.; Cortes, C.; Lawrence, N.; Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
26. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
27. Huang, G.B.; Ramesh, M.; Berg, T.; Learned-Miller, E. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*; Technical Report 07-49; University of Massachusetts: Amherst, MA, USA, 2007.
28. Kazemi, V.; Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE/CFV Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1867–1874.
29. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep Learning Face Attributes in the Wild. In Proceedings of the International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
30. Lee, C.H.; Liu, Z.; Wu, L.; Luo, P. MaskGAN: Towards Diverse and Interactive Facial Image Manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
31. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
32. Pasquini, C.; Amerini, I.; Boato, G. Media forensics on social media platforms: A survey. *EURASIP J. Inf. Secur.* **2021**, *2021*, 1–19. [CrossRef]
33. Boato, G.; Pasquini, C.; Stefani, A.; Verde, S.; Miorandi, D. TrueFace: A dataset for the detection of synthetic face images from social networks. In Proceedings of the IEEE/IAPR International Joint Conference on Biometrics, Abu Dhabi, United Arab Emirates, 10–13 October 2022.
34. Karras, T.; Aittala, M.; Laine, S.; Härkönen, E.; Hellsten, J.; Lehtinen, J.; Aila, T. Alias-Free Generative Adversarial Networks. In Proceedings of the NeurIPS, Virtual, 13 December 2021.
35. Chan, E.R.; Lin, C.Z.; Chan, M.A.; Nagano, K.; Pan, B.; Mello, S.D.; Gallo, O.; Guibas, L.; Tremblay, J.; Khamis, S.; et al. Efficient Geometry-aware 3D Generative Adversarial Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–20 June 2022.
36. Bińkowski, M.; Donahue, J.; Dieleman, S.; Clark, A.; Elsen, E.; Casagrande, N.; Cobo, L.C. High Fidelity Speech Synthesis with Adversarial Networks. In Proceedings of the ICLR, Addis Ababa, Ethiopia, 26–30 April 2020.
37. Xu, J.; Sun, X.; Ren, X.; Lin, J.; Wei, B.; Li, W. DP-GAN: Diversity-Promoting Generative Adversarial Network for Generating Informative and Diversified Text. *arXiv* **2018**, arXiv:1802.01345.
38. Bhavsar, K.; Vakharia, V.; Chaudhari, R.; Vora, J.; Pimenov, D.Y.; Giasin, K. A Comparative Study to Predict Bearing Degradation Using Discrete Wavelet Transform (DWT), Tabular Generative Adversarial Networks (TGAN) and Machine Learning Models. *Machines* **2022**, *10*, 176. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.