DeepFake Detection by Analyzing Convolutional Traces

Luca Guarnera
University of Catania - iCTLab
Catania, Italy
luca.guarnera@unict.it

Oliver Giudice
University of Catania
Catania, Italy
giudice@dmi.unict.it

Sebastiano Battiato
University of Catania - iCTLab
Catania, Italy
battiato@dmi.unict.it

Abstract

The Deepfake phenomenon has become very popular nowadays thanks to the possibility to create incredibly realistic images using deep learning tools, based mainly on ad-hoc Generative Adversarial Networks (GAN). In this work we focus on the analysis of Deepfakes of human faces with the objective of creating a new detection method able to detect a forensics trace hidden in images: a sort of fingerprint left in the image generation process. The proposed technique, by means of an Expectation Maximization (EM) algorithm, extracts a set of local features specifically addressed to model the underlying convolutional generative process. Ad-hoc validation has been employed through experimental tests with naive classifiers on five different architectures (GDWCT, STARGAN, ATTGAN, STYLEGAN, STYLEGAN2) against the CELEBA dataset as ground-truth for non-fakes. Results demonstrated the effectiveness of the technique in distinguishing the different architectures and the corresponding generation process.

1. Introduction

One of the phenomena that is rapidly growing is the well-known Deepfake: the possibility to automatically generate and/or alter/swap a person's face in images and videos using algorithms based on *Deep Learning* technology. It is possible to generate excellent results by creating new multimedia contents that cannot be easily recognized as real or fake by human eye. Then, the term Deepfake refers to all those multimedia contents synthetically altered or created by means of machine learning generative models.

Various examples of Deepfake, involving celebrities, are easily discoverable on the web: the insertion of Nicholas Cage [1] in movies where he did not act like "Fight Club" and "The Matrix" or the impressive video in which Jim Carrey [2] plays Shining in place of Jack Nicholson. Other more worrying examples are the video of ex US President Barack Obama (Figure 1(a)), created by Buzzfeed [3] in collaboration with Monkeypaw Studios, or the video in which Mark Zuckerberg [4] (Figure 1(b)) claims a series of statements about his platform ability to steal users' data. Even in Italy, in September 2019, the satirical TV program "Striscia La Notizia" [5] showed a video of the ex-premier Matteo Renzi talking about his colleagues in a "not so respectful" way (Figure 1 (c)). Indeed, Deepfakes may have serious repercussions on the authenticity of the news spread by the mass-media while representing a new threat for politics, companies and individual privacy. In this dangerous scenario, tools are needed to unmask the Deepfakes or just detect them.

Several big companies have decided to take action against this phenomenon: Google has created a database of fake videos [36] to support researchers who are developing new techniques to detect them, while Facebook and Microsoft have launched the Deepfake Detection Challenge initiative [6].

In this paper a new Deepfake detection method will be introduced focused on images representing human faces. At first an Expectation Maximization (EM) algorithm [29], extracts a set of local features specifically addressed to model the convolutional traces that could be found in images. Then, naive classifiers were trained to discriminate between authentic images and images generated by the five most realistic architectures as today (GDWCT, STARGAN, ATTGAN, STYLEGAN, STYLEGAN2). Experimental results demonstrated that the information modelled by EM is related to the specific architecture that generated the image thus giving the overall detection solution explainability, being also of great value for forensic investigations (e.g., camera model identification techniques of image forensics). Moreover, a multitude of experiments will be presented not

[1] https://www.youtube.com/watch?v=-yQxsIWO2ic
[2] https://www.youtube.com/watch?v=Dx59bskG8dc
[3] https://www.youtube.com/watch?v=cQ54GDm1eL0
[4] https://www.youtube.com/watch?v=NbedWhzx1rs
[5] https://www.striscialanotizia.mediaset.it/video/il-fuorionda-di-matteo-renzi_59895.shtml
[6] https://deepfakedetectionchallenge.ai/

Figure 1. Examples of Deepfake: (a) Obama (Buzzfeed in collaboration with Monkeypaw Studios); (b) Mark Zuckerberg (Bill Posters and Daniel Howe in partnership with advertising company Canny); (c) Matteo Renzi (the italian TV program "Striscia la Notizia").

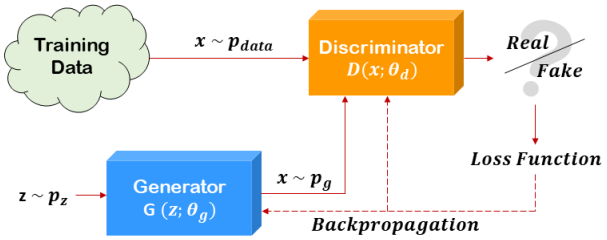

Figure 2. Simplified description of a GAN learning framework.

only to demonstrated the effectiveness of the technique but also to demonstrate to be un-comparable with state-of-the-art: tests were carried out on an almost-in-the-wild dataset with images generated by five different techniques with different image sizes. As today, all proposed technique work with specific image sizes and against at most one GAN technique.

The remainder of this paper is organized as follows: Section 2 presents some Deepfake generation and detection methods. The proposed detection technique is explained in Section 3 as regards the feature extraction phase while the classification phase and experimental results are reported in Section 4. Finally, Section 5 concludes the paper with insights for future works.

2. Related Works

Deepfakes are generally created by techniques based on Generative Adversarial Networks (GANs) firstly introduced by Goodfellow et al. [14]. Authors proposed a new framework for estimating generative models via an adversarial mode in which two models simultaneously train: a generative model G, that captures the data distribution, and a discriminative model D, able to estimate the probability that a sample comes from the training data rather than from G. The training procedure for G is to maximize the probability of D making a mistake thus resulting to a min-max two-player game. Mathematically, the generator accepts a random input z with density p_z and returns an output $x = G(z, \Theta_g)$ according to a certain probability distribution p_g (Θ_g represent the parameters of the generative model). The discriminator, $D(x, \Theta_d)$ computes the probability that x comes from the distribution of training data p_{data} (Θ_d represents the parameters of the discriminative model). The overall objective is to obtain a generator, after the training phase, which is a good estimator of p_{data}. When this happens, the discriminator is "deceived" and will no longer be able to distinguish the samples from p_{data} and p_g; therefore p_g will follow the targeted probability distribution, i.e. p_{data}. Figure 2 shows a simplified description of a GAN framework. In the case of Deepfakes, G can be thought as a team of counterfeiters trying to produce fake currency, while D stands to the police, trying to detect the malicious activity. G and D can be implemented as any kind of generative model, in particular when deep neural networks are employed results become extremely accurate. Through recent years, many GAN architectures were proposed for different applications e.g., image to image translation [45], image super resolution [24], image completion [17], and text-to-image generation [35].

2.1. Deepfake Generation Techniques

An overview on Media forensics with particular focus on Deepfakes has been recently proposed in [41].

STARGAN is a method capable of performing image-to-image translations on multiple domains using a single model. Proposed by Choi et al. [6] was trained on two different types of face datasets: CELEBA [27] containing 40 labels related to facial attributes such as hair color, gender and age, and RaFD dataset [22] containing 8 labels corresponding to different types of facial expressions ("happy", "sad", etc.). Given a random label as input, such as hair color, facial expression, etc., STARGAN is able to perform an image-to-image translation operation. Results have been compared with other existing methods [26, 30, 45] and showed how STARGAN manages to generate images of superior visual quality.

Style Generative Adversarial Network, namely STYLEGAN [19], changed the generator model of STARGAN by means of mapping points in latent space to an intermediate latent space which controls the *style* output at each point of the generation process. Moreover the introduction of noise

as a source of variation in those mentioned points demonstrates to achieve better results. Thus, STYLEGAN is capable not only of generating impressively photorealistic and high-quality photos of faces, but also offers control parameters in terms of the overall *style* the generated image at different levels of detail. While being able to create realistic pseudo-portraits, small details might reveal the fakeness of generated images. To correct those imperfections in STYLEGAN, Karras et al. made some improvements to the generator (including re-designed normalization, multi-resolution, and regularization methods) proposing STYLEGAN2 [20].

Instead of imposing constraints on latent representation, He et al. [15], proposed a new technique called ATTGAN in which an attribute classification constraint is applied to the generated image, in order to guarantee only the correct modifications of the desired attributes. The authors used CELEBA [27] and LFW [16] datasets, and performed various tests comparing ATTGAN with VAE/GAN [23], IcGAN [30] and STARGAN [6], Fader Networks [21], Shen et al. [37] and CycleGAN [45]. Achieved results showed that ATTGAN exceeds the state of the art on the realistic modification of facial attributes.

The latter style transfer approach worth to be mentioned is the work of Cho et al. [5], where they propose a group-wise deep whitening-and coloring method (GDWCT) for a better styling capacity. They used CELEBA [27], Artworks [45], cat2dog [25], Ink pen and watercolor classes from Behance Artistic Media (BAM) [43], and Yosemite datasets [45] as dataset. GDWCT has been compared with various cutting-edge methods in image translation and style transfer improving not only computational efficiency but also quality of generated images.

In this paper, the five most famous and effective architectures in state-of-the-art for face Deepfakes were taken into account: STARGAN [6], STYLEGAN [19], STYLEGAN2 [20], ATTGAN [15] and GDWCT [5]. As described above, they are different in goals and structure. Table 1 resumes the differences of the techniques in terms of image size, dataset and type of input, goal and architecture structure.

2.2. Deepfake detection methods

Being able to understand if an image is the result of a generative Neural Network process turns out to be a complicated problem, even for human eyes. However, the problem of authenticating an image (or specifically a digital image) is not new [2, 31, 38]. Many works try to reconstruct the history of an image[13]; others try to identify the anomalies, such as the study on the analysis of interpolation effects through CFA (Color Filtering Array) [32], analyzing compression parameters [3, 11, 12], etc. Given the peculiarity of Deepfakes, state-of-the-art image analysis methods tend to fail and more refined ones are needed.

Thanks to a new discriminator that uses "contrastive loss" it is possible to find the typical characteristics of the synthesized images generated by different GANs and therefore detect such fake images by means of a classifier. Rossler et al. [36] proposed an automated benchmark for fake detection, based mainly on four manipulation methods: two computer graphics-based methods (Face2Face [40], FaceSwap [7]) and 2 learning-based approaches (DeepFakes [8], NeuralTextures [39]). They addressed the problem of fake detection as a binary classification problem for each frame of manipulated videos, considering different techniques present in the state of the art [1, 4, 7, 8, 10, 34].

Zhang et al. [44] proposed a method to classify Deepfakes considering the spectra of the frequency domain as input. The authors proposed a GAN simulation framework, called AutoGAN, in order to emulate the process commonly shared by popular GAN models. Results obtained by the authors achieved very good performances in terms of binary classification between authentic and fake images. Also Durall et al. [9] presented a method for Deepfakes detection based on the analysis in the frequency domain. The authors combined high-resolution authentic face images from different public datasets (CELEBA-HQ data set [18], Flickr-Faces-HQ data set [19]) with fakes (100K Faces project [9], this person does not exist [10]), creating a new dataset called Faces-HQ. By means of naive classifiers they obtained good results in terms of overall accuracy.

Differently from described approaches, in this paper the possibility to capture the underlying traces of a possible Deepfake is investigated by employing a sort of reverse engineering of the last computational layer of a given GAN architecture. This method will give explainability to the predictions of Deepfakes being of great value for forensic investigations: not only it is able to classify an image as fake but also can predict the most probable technique used for generation being in this way similar to camera model detection in image forensics analysis [2]. The underlying idea of the technique is to find the main periodic components (e.g. transpose computational layer) on generated images. A similar strategy was proposed some time ago in a seminal paper of Popescu et al. [32] devoted to point out the presence of digital forgeries in CFA interpolated images. Another difference from state-of-the-art is the working scenario: the proposed technique demonstrates to achieve good results in a almost-in-the-wild scenario with images generated by five different techniques and image sizes.

[7] https://github.com/MarekKowalski/FaceSwap/
[8] https://github.com/deepfakes/faceswap/
[9] https://generated.photos/
[10] https://thispersondoesnotexist.com/

Method	Number of images generated	Size	Data input to the network	Goal of the network	Kernel size of the latest Convolution Layer
GDWCT [5]	3369	216x216	CELEBA	Improves the styling capability	4x4
STARGAN [6]	5648	256x256	CELEBA	Image-to-image translations on multiple domains using a single model	7x7
ATTGAN [15]	6005	256x256	CELEBA	Transfer of face attributes with classification constraints	4x4
STYLEGAN [19]	9999	1024x1024	CELEBA-HQ FFHQ	Transfer semantic content from a source domain to a target domain characterized by a different style	3x3
STYLEGAN2 [20]	3000	1024x1024	FFHQ	Transfer semantic content from a source domain to a target domain characterized by a different style	3x3

Table 1. Details of Deepfake GAN architectures employed for analysis. For each one is reported: all images generated, the generated image sizes, the original input used to train the neural network, the goal of the network and the kernel size of last convolutional layer.

3. Extracting Convolutional Traces

The most common and effective technical solutions able to generate Deepfakes are the Generative Adversarial Networks specifically deep ones. For all the techniques described before, the generator G is composed of Transpose Convolution layers [33]. In Neural Networks like CNNs, Convolution operations apply a filter, namely kernel, to the input multidimensional array. After each convolution layer a pooling operation is needed to reduce output dimensional size w.r.t. input. On the other hand, in generative models the Transpose Convolution Layers are employed. They also apply kernels to input but they act inversely in order to obtain an output larger but proportional to the input dimensions.

The starting idea of the proposed approach is that local correlation of pixels in Deepfakes are dependent exclusively on the operations performed by all the layers present in the GAN which generate it; specifically the (latter) transpose convolution layers. In order to find these trace, unsupervised machine learning techniques were taken into account. Indeed, different unsupervised learning techniques aim at creating clusters containing instances of the input dataset with high similarity between instances of the same cluster while having high dissimilarity between instances belonging to different clusters. These clusters can represent the "hidden" structure of the dataset analyzed. Therefore, the clustering technique must estimate which are the parameters of the distributions that most likely generated the training samples. Based on this principle, an Expectation Maximization (EM) algorithm [29] was employed in order to define a conceptual mathematical model able to capture the pixel correlation of the images (e.g. spatially). The result of EM is a feature vector representing the structure of the Transpose Convolution Layers employed during the generation of the image, encoding in some sense is such images if a Deepfake or not.

The initial goal is to extract a description, from input image I, able to numerically represent the local correlations between each pixel in a neighbourhood. This can be done by means of convolution with a kernel k of $N \times N$ size:

$$I[x,y] = \sum_{s,t=-\alpha}^{\alpha} k_{s,t} * I[x+s, y+t] \qquad (1)$$

In Equation 1, the value of the pixel $I[x,y]$ is computed considering a neighborhood of size $N \times N$ of the input data. It is clear that the new estimated information $I[x,y]$ mainly depends on the kernel used in the convolution operation, which establishes a mathematical relationship between the pixels. For this reason, our goal is to define a vector k of size $N \times N$ able to capture this hidden and implicit relationship which characterize of forensic trace we want to exploit.

Let's assume that the element $I[x,y]$ belongs to one of the following models:

- M_1: when the element $I[x,y]$ satisfies Equation 1;
- M_2: otherwise.

The EM algorithm is employed with its two different steps:

1. **Expectation step**: computes the (density of) probability that each element belongs to model (M_1 or M_2);
2. **Maximization step**: estimates the (weighted) parameters based on the probabilities of belonging to instances of (M_1 or M_2).

Let's suppose that M_1 and M_2 have different probability distributions with M_1 Gaussian distribution with zero mean and unknown variance and M_2 uniform. In the Expectation

step, the Bayes rule that $I[x,y]$ belongs to the model M_1 is computed as follows:

$$Pr\{I[x,y] \in M_1 \mid I[x,y]\} = \\ = \frac{Pr\{I[x,y] \mid I[x,y] \in M_1\} * Pr\{I[x,y] \in M_1\}}{\sum_{i=1}^{2} Pr\{I[x,y] \mid I[x,y] \in M_i\} * Pr\{I[x,y] \in M_i\}} \tag{2}$$

where the probability distribution of M_1 which represents the probability of observing a sample $I[x,y]$, knowing that it was generated by the model M_1 is:

$$Pr\{I[x,y] \mid I[x,y] \in M_1\} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(R[x,y])^2}{2\sigma^2}} \tag{3}$$

where

$$R[x,y] = \left| I[x,y] - \sum_{s,t=-\alpha}^{\alpha} k_{s,t} I[x+s, y+t] \right| \tag{4}$$

.

The variance value σ^2, which is still unknown, is then estimated in the Maximization step. Once defined if $I[x,y]$ belongs to model M_1 (or M_2), the values of the vector $\vec{k}$ are estimated using least squares method, minimizing the following:

$$E(\vec{k}) = \sum_{x,y} w[x,y] \left(I[x,y] - \sum_{s,t=-\alpha}^{\alpha} k_{s,t} I[x+s, y+t] \right)^2 \tag{5}$$

where $w \equiv Pr\{I[x,y] \in M_1 \mid I[x,y]\}$ (2). This error function (5) can be minimized by computing the gradient of vector $\vec{k}$. The update of $k_{i,j}$ is carried out by computing the partial derivative of (5) as follows:

$$\frac{\partial E}{\partial k_{i,j}} = 0 \tag{6}$$

Hence, the following linear equations system is obtained:

$$\sum_{s,t=-\alpha}^{\alpha} k_{s,t} \left(\sum_{x,y} w[x,y] I[x+i, y+j] I[x+s, y+t] \right) = \\ = \sum_{x,y} w[x,y] I[x+i, y+j] I[x,y] \tag{7}$$

The two steps of the EM algorithm are iteratively repeated. A pseudo-code description is provided in *Algorithm 1:Expectation-Maximization*. The algorithm is applied to each channel of the input image (RGB color space).

Algorithm 1: Expectation-Maximization Algorithm

Data: Image I
Result: $\vec{k}$
Initialize N //Kernel size
Initialize σ_0
Set $\vec{k}$ random of size NxN
Set R, P, W matrices with 0 values of the same size as I
Set p_0 as 1/size of the range of values of I
for $n = 1;\ n < 100\ n+ = 1$ **do**
  //Expectation Step
  for $\forall$ *values in* I **do**
    $R[x,y] = \left| I[x,y] - \sum_{s,t=-\alpha}^{\alpha} k_{s,t} I[x+s, y+t] \right|$
    $P[x,y] = \frac{1}{\sigma_n\sqrt{2\pi}} e^{-\frac{R[x,y]}{2\sigma_n^2}}$
    $W[x,y] = \frac{P[x,y]}{P[x,y]+p_0}$
  //Maximization Step
  Calculate $k_{s,t}^{(n+1)}$ as shown in the formula 7

The obtained feature vector $\vec{k}$, has dimensions dependent to parameter α. Note that the element $k_{0,0}$ will always be set equal to 0 ($k_{0,0} = 0$). Thus, for example, if a kernel k with 3×3 size is employed, the resulting $\vec{k}$ will be a vector of 24 elements (since the values $k_{0,0}$ are excluded). This is obtained by concatenating the features extracted from each of the three RGB channels.

The computational complexity of the EM algorithm can be estimated to be linear in d (the number of characteristics of the input data taken into consideration), n (the number of objects) and t (the number of iterations).

4. Classification Phase and Results

Six datasets of images were taken into account for training and testing purposes: one containing only authentic face images of celebrities (CELEBA), and the others containing DeepFakes generated by five different GANs (STARGAN, STYLEGAN, STYLEGAN2, GDWCT, ATTGAN). For STYLEGAN and STYLEGAN2, images were downloaded from STYLEGAN [11] and STYLEGAN2 [12] respectively; while STARGAN, ATTGAN and GDWCT were employed in inference mode to generate their respective image datasets. An overview of the DeepFake data generated for each GAN is reported in the Table 1.

[11] https://drive.google.com/drive/folders/1uka3a1noXHAydRPRbknqwKVGODvnmUBX

[12] https://drive.google.com/drive/folders/1QHc-yF5C3DChRwSdZKcx1w6K8JvSxQi7

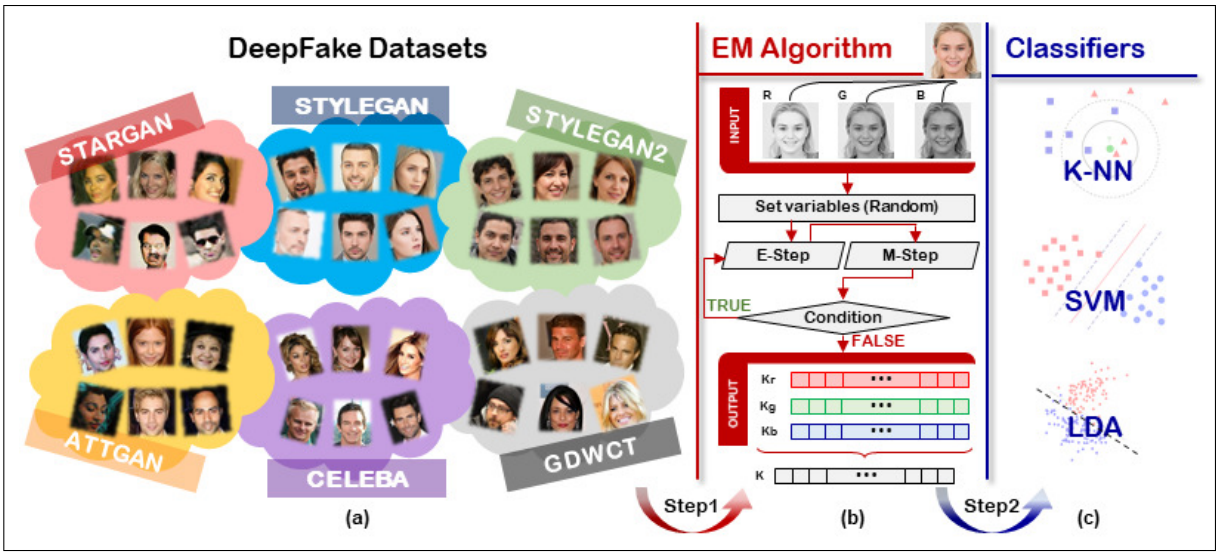

Figure 3. Overall pipeline. (a) Datasets of real (CELEBA) and Deepfake images, (b) For each images in (a) features are extracted by means of EM algorithm; (c) types of classifiers used (K-NN, SVM, LDA).

The EM algorithm, as described in previous Section, was employed on the 6 datasets described above, in order to extract a feature vector $\vec{k}$ able to describe the convolutional traces left in images. EM was employed with kernels of increasing sizes ($3, 4, 5$ and 7) [13]. The obtained feature vector was employed as input of different naive classifiers (K-NN, SVM and LDA) with different tasks: (i) discriminating authentic image from one specific GAN and (ii) discriminating authentic images from Deepfakes. The overall classification pipeline of the proposed approach is briefly summarized in Figure 3. Let's first analyse the discriminative power of the extracted feature vector in order to distinguish authentic images (CELEBA) from each of the considered GANs (CELEBA Vs STARGAN, CELEBA Vs STYLEGAN, CELEBA Vs STYLEGAN2, CELEBA Vs ATTGAN, CELEBA Vs GDWCT). Figure 4 shows a visible representation by means of t-SNE [28]: in which it is possible to notice, how some categories of networks that create Deepfake can be "linearly" separable from authentic samples. However in most case the separation is utterly clear.

Classification tests were carried out on the obtained feature vectors with, as expected from what seen from t-SNE representation, excellent results. All the classification results are reported in Table 2. In particular, it is possible to note that:

- **CELEBA Vs ATTGAN** the maximum classification accuracy of **92.67%**, was obtained with KNN - K = 3, and kernel size of $3x3$.
- **CELEBA Vs GDWCT**: the maximum classification accuracy of **88.40%**, was obtained with KNN - K = 3,5,7, and kernel size of $3x3$.
- **CELEBA Vs STARGAN**: the maximum classification accuracy of **93.17%**, was obtained with linear SVM, and kernel size of $7x7$.
- **CELEBA Vs STYLEGAN**: the maximum classification accuracy of **99.65%**, was obtained with KNN - K = 3,5,7,9, and kernel size of $4x4$.
- **CELEBA Vs STYLEGAN2**: the maximum classification accuracy of **99.81%**, was obtained with linear SVM, and kernel size of $4x4$.

The kernel size used by convolution layers in the neural networks represents one of the elements to identify the forensic trace that we are looking for. Table 1 shows the kernel size (and other information) of the neural networks that we have taken into account for our experiments.

As described above, the structure of the GAN plays a fundamental role in the Deepfakes detection, in particular for what regards the generator structure. Considering the images from STYLEGAN and STYLEGAN2, it is possible to distinguish them, as the authors of the STYLEGAN2 architecture have only updated parts of the generator in order to remove some imperfections of STYLEGAN. This further confirms the hypothesis, since even a slight modification of the generator, in particular to the convolution layers, leaves different traces in the images generated. When trying to distinguish the images from STYLEGAN with those of

[13] Typical kernel size used by the latest Transpose Convolution Layers (which have a fundamental role in the creation of the Deepfake images) of the different GAN architectures

	CELEBA Vs ATTGAN				CELEBA Vs GDWCT				CELEBA Vs STARGAN				CELEBA Vs STYLEGAN				CELEBA Vs STYLEGAN2			
	Kernel Size				Kernel Size				Kernel Size				Kernel Size				Kernel Size			
	3x3	4x4	5x5	7x7	3x3	4x4	5x5	7x7	3x3	4x4	5x5	7x7	3x3	4x4	5x5	7x7	3x3	4x4	5x5	7x7
3-NN	**92.67**	86.50	84.50	85.33	**88.40**	73.17	73.00	74.33	**90.50**	89.00	88.67	85.17	93.00	**99.65**	98.26	99.55	96.99	**99.61**	98.75	97.77
5-NN	**92.00**	86.50	84.83	86.17	**88.40**	75.67	74.17	76.67	**88.83**	**88.83**	88.17	85.00	93.00	**99.65**	98.26	99.32	97.39	**99.61**	98.21	97.55
7-NN	**91.00**	87.67	85.33	85.67	**88.40**	76.67	71.33	78.67	**89.33**	89.17	88.00	84.83	93.50	**99.65**	98.07	99.09	97.39	**99.42**	98.21	97.55
9-NN	**90.83**	87.67	84.83	86.50	**87.70**	76.83	71.17	79.00	**89.33**	89.17	87.50	84.67	92.83	**99.65**	98.07	99.32	97.19	**99.42**	98.39	97.10
11-NN	**91.00**	86.83	85.33	85.83	**88.05**	76.67	72.83	77.00	**89.17**	88.67	86.67	83.50	93.17	**99.48**	98.07	99.32	96.99	**99.42**	97.85	97.10
13-NN	**91.00**	87.17	84.50	85.33	**87.87**	75.33	73.50	77.17	88.33	**89.33**	87.50	83.50	93.50	**99.48**	98.07	99.09	97.39	**99.22**	97.67	97.10
SVM	**90.50**	89.67	90.33	87.00	**87.35**	76.50	79.00	80.50	90.00	88.50	88.83	**93.17**	92.00	98.96	**99.42**	98.41	96.99	**99.81**	99.46	97.77
LDA	**89.50**	88.50	**89.50**	87.17	**87.52**	76.00	79.33	81.67	89.67	87.83	88.83	**90.00**	92.50	**99.31**	98.84	99.09	96.79	**99.61**	99.10	97.77

Table 2. Overall accuracy between CELEBA vs. each one of the considered GANs. Results are presented w.r.t. all the different kernel sizes ($3x3$, $4x4$, $5x5$, $7x7$) and with different classifiers: KNN, with $k \in \{3, 5, 7, 9, 11, 13\}$; Linear SVM, Linear Discriminant Analysis (LDA).

	CELEBA Vs DeepNetworks			
	Kernel Size			
	3x3	4x4	5x5	7x7
3-NN	**89.96**	84.90	80.76	82.69
5-NN	**90.22**	86.63	82.48	82.77
7-NN	**89.57**	87.12	82.48	84.27
9-NN	**89.51**	86.73	83.31	84.27
11-NN	**89.25**	87.21	83.69	83.97
13-NN	**89.57**	87.31	84.20	83.45
SVMLinear	88.02	**88.75**	86.05	85.85
SVMsigmoid	**86.08**	72.60	83.38	63.66
SVMrbf	**89.77**	89.71	86.24	87.43
SVMPoly	82.51	86.06	84.65	**86.61**
LDA	87.56	**88.65**	86.11	85.48

Table 3. Overall accuracy between CELEBA with all Deep Neural Network, with different kernel size ($3x3$, $4x4$, $5x5$, $7x7$ - obtained through the EM algorithm) and with different classifiers used: KNN, with $k \in \{3, 5, 7, 9, 11, 13\}$; SVM (linear, sigmoid, rbf, polynomial), Linear Discriminant Analysis (LDA).

	STYLEGAN Vs STYLEGAN2			
	Kernel Size			
	3x3	4x4	5x5	7x7
3-NN	89.36	83.57	**90.51**	87.24
5-NN	89.56	86.41	**89.87**	85.52
7-NN	89.16	85.40	**90.93**	87.59
9-NN	88.55	83.98	**89.87**	87.93
11-NN	88.35	83.37	**90.30**	87.24
13-NN	89.36	82.76	**89.66**	87.93
SVM	91.77	95.13	99.16	**99.31**
LDA	91.16	94.52	**98.73**	98.28

Table 4. Overall accuracy between STYLEGAN and STYLEGAN2, with all the different kernel size ($3x3$, $4x4$, $5x5$, $7x7$ - obtained through the EM algorithm) and with different employed classifiers: KNN, with $k \in \{3, 5, 7, 9, 11, 13\}$; Linear SVM, Linear Discriminant Analysis (LDA).

STYLEGAN2, we get a maximum accuracy of the **99.31%** (Table 4).

Finally, another type of classification was the comparison between CELEBA original images and all the images generated with all the networks as a binary classification problem. In this test, a further analysis of the two-dimensional t-SNE was carried out. Figure 5) shows that, in this case, samples cannot be linearly separated. For this reason, other non-linear classifiers were taken into account reaching a maximum accuracy of **90.22**% (with KNN, K=5), with kernel employed in the EM of size 3×3. Table 3 shows the obtained results in the binary classification task.

Many additional experiments were carried out to furtherly demonstrate the effectiveness of the extracted feature vector as a descriptor of the hidden convolutional trace. Specifically results w.r.t. classification tests between different combinations of GANs are described furtherly conferming the robustness of the technique. Also other t-SNE representations are provided and can be found at the following address https://iplab.dmi.unict.it/mfs/DeepFake/.

Finally, it is worth to point out that during the research activity a deep neural network technique was employed to detect Deepfakes on the datasets described above. Tests carried out with VGG-16[14] on both spatial and frequency domain of images achieved a best result of 53% of accuracy in the binary classification task showing that a deep learning approach is not able to extract what the proposed approach was able to. Our results are similar in terms of overall performance by experiments exploited in Wang et al. [42] that is actually able to reach very high results by simply using a discriminator trained on one family of GANs and using it to infer if images are real or generated from other types of GANs.

5. Conclusions and future works

The final result of our study to counter the Deepfake phenomenon was the creation of a new detection method based on features extracted through the EM algorithm. The underlying fingerprint has been proven to be effective to discriminate between images generated by recent GANs architectures specifically devoted to generate realistic people's face. Some more works will be devoted to investigate the role of the kernel dimensions. Also the possibility to extend such

[14] https://github.com/1297rohit/VGG16-In-Keras

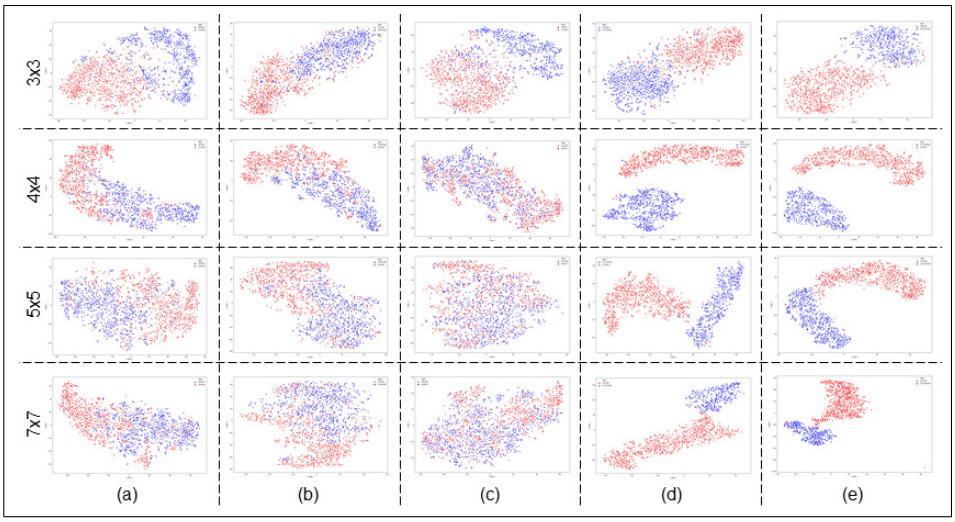

Figure 4. Two-dimensional t-SNE representation (CELEBA: red; DeepNetwork: blue) of all kernel sizes for each classification task: (a) CELEBA – ATTGAN; (b) CELEBA – STARGAN; (c) CELEBA – GDWCT; (d) CELEBA – STYLEGAN; (e) CELEBA – STYLEGAN2.

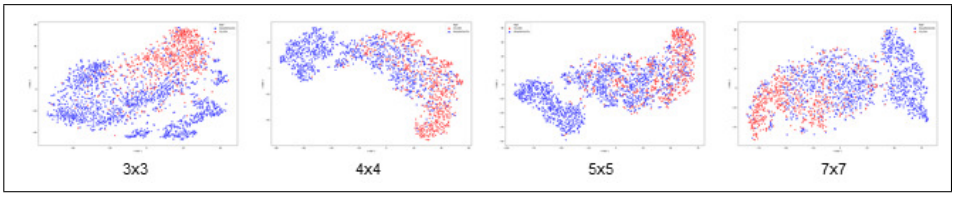

Figure 5. Two-dimensional t-SNE representation (CELEBA: red; DeepNetwork: blue) of a binary classification problem (with different kernel size): CELEBA Vs DeepNetworks.

methodology to video's analysis and/or evaluate the robustness with respect to standard image editing (e.g. photometric and compression) and malicious processing (e.g. antiforensics) devoted to mask the underlying forensic traces will be considered. In general one of the key aspect will the possibility to adapt the method in situations on the "wild" without any a-priori knowledge of the generation process.

Acknowledgement

This research was supported by iCTLab s.r.l. - Spin-off of University of Catania (https://www.ictlab.srl), which provided domain expertise and computational power that greatly assisted the activity.

References

[1] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7. IEEE, 2018. 3

[2] S. Battiato, O. Giudice, and A. Paratore. Multimedia forensics: discovering the history of multimedia contents. In *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, pages 5–16. ACM, 2016. 3

[3] S. Battiato and G. Messina. Digital forgery estimation into DCT domain: a critical analysis. In *Proceedings of the First ACM Workshop on Multimedia in Forensics*, pages 37–42, 2009. 3

[4] B. Bayar and M. C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2016. 3

[5] W. Cho, S. Choi, D. K. Park, I. Shin, and J. Choo. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10639–10647, 2019. 3, 4

[6] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8789–8797, 2018. 2, 3, 4

[7] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017. 3

[8] D. Cozzolino, G. Poggi, and L. Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, pages 159–164, 2017. 3

[9] R. Durall, M. Keuper, F. Pfreundt, and J. Keuper. Unmasking deepfakes with simple features. *arXiv preprint arXiv:1911.00686*, 2019. 3

[10] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, 2012. 3

[11] F. Galvan, G. Puglisi, A. R. Bruna, and S. Battiato. First quantization matrix estimation from double compressed JPEG images. *IEEE Transactions on Information Forensics and Security*, 9(8):1299–1310, 2014. 3

[12] O. Giudice, F. Guarnera, A. Paratore, and S. Battiato. 1-D DCT domain analysis for JPEG double compression detection. In *Proceeedings of International Conference on Image Analysis and Processing*, pages 716–726. Springer, 2019. 3

[13] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato. *A Classification Engine for Image Ballistics of Social Data*, pages 625–636. Springer International Publishing, 2017. 3

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[15] Z. He, W. Zuo, M. Kan, S. Shan, and X. Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE Transactions on Image Processing*, 28(11):5464–5478, 2019. 3, 4

[16] G. B Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition, Oct 2008, Marseille, France. inria-00321923*, 2008. 3

[17] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017. 2

[18] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 3

[19] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2, 3, 4

[20] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. *arXiv preprint arXiv:1912.04958*, 2019. 3, 4

[21] G. Lample, N. Zeghidour, N. Usunier, A. Bordes, L. Denoyer, and M. Ranzato. Fader networks: Manipulating images by sliding attributes. In *Advances in Neural Information Processing Systems*, pages 5967–5976, 2017. 3

[22] O. Langner, R. Dotsch, G. Bijlstra, D. HJ Wigboldus, S. T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388, 2010. 2

[23] A. Boesen Lindbo Larsen, S. Kaae Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015. 3

[24] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017. 2

[25] H. Lee, H. Tseng, J. Huang, M. Singh, and M. Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–51, 2018. 3

[26] M. Li, W. Zuo, and D. Zhang. Deep identity-aware transfer of facial attributes. *arXiv preprint arXiv:1610.05586*, 2016. 2

[27] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 2, 3

[28] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. 6

[29] T. K Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996. 1, 4

[30] G. Perarnau, J. Van De Weijer, B. Raducanu, and J. M Álvarez. Invertible conditional GANs for image editing. *arXiv preprint arXiv:1611.06355*, 2016. 2, 3

[31] A. Piva. An overview on image forensics. *ISRN Signal Processing*, 2013:22, 2013. 3

[32] A. C Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *IEEE Transactions on Signal Processing*, 53(10):3948–3959, 2005. 3

[33] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 4

[34] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2017. 3

[35] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 2

[36] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–11, 2019. 1, 3

[37] W. Shen and R. Liu. Learning residual images for face attribute manipulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4030–4038, 2017. 3

[38] M. C. Stamm, M. Wu, and K. J. R. Liu. Information forensics: An overview of the first decade. *IEEE Access*, 1:167–200, 2013. 3

[39] J. Thies, M. Zollhöfer, and M. Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[40] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of RGB videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 3

[41] L. Verdoliva. Media forensics and deepfakes: an overview. *arXiv preprint arXiv:2001.06564*, 2020. 2

[42] S. Wang, O. Wang, R. Zhang, A. Owens, and A. Efros. Cnn-generated images are surprisingly easy to spot...for now. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,*, 2020. 7

[43] M. J Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie. Bam! the behance artistic media dataset for recognition beyond photography. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1202–1211, 2017. 3

[44] X. Zhang, S. Karaman, and S. Chang. Detecting and simulating artifacts in GAN fake images. *arXiv preprint arXiv:1907.06515*, 2019. 3

[45] J. Zhu, T. Park, P. Isola, and A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 2, 3