Benford's law feature extraction plus random forest is used. Outperforms Xception plus steganalysis and SVM

On the use of Benford's law to detect GAN-generated images

Nicolò Bonettini
DEIB
Politecnico di Milano
Milano, Italy
nicolo.bonettini@polimi.it

Paolo Bestagini
DEIB
Politecnico di Milano
Milano, Italy
paolo.bestagini@polimi.it

Simone Milani
Department of Information Engineering
University of Padova
Padova, Italy
simone.milani@dei.unipd.it

Stefano Tubaro
DEIB
Politecnico di Milano
Milano, Italy
stefano.tubaro@polimi.it

***Abstract*—The advent of Generative Adversarial Network (GAN) architectures has given anyone the ability of generating incredibly realistic synthetic imagery. The malicious diffusion of GAN-generated images may lead to serious social and political consequences (e.g., fake news spreading, opinion formation, etc.). It is therefore important to regulate the widespread distribution of synthetic imagery by developing solutions able to detect them. In this paper, we study the possibility of using Benford's law to discriminate GAN-generated images from natural photographs. Benford's law describes the distribution of the most significant digit for quantized Discrete Cosine Transform (DCT) coefficients. Extending and generalizing this property, we show that it is possible to extract a compact feature vector from an image. This feature vector can be fed to an extremely simple classifier for GAN-generated image detection purpose.**

***Index Terms*—image forensics, GAN, Benford's law**

I. Introduction

With the advent of modern deep learning solutions such as GANs, a new series of image and video editing tools has been made available to everyone (e.g., Recycle-GAN [1], StyleGAN [2], etc.). These techniques allow to synthesize realistic and visually-pleasant artificial images not resorting to complex Computer-Generated Imagery (CGI) techniques required in the past. Unfortunately, this great step forward in technology came at a price. Indeed, GANs can be maliciously used by everyone to generate very realistic image forgeries to manipulate people's opinion through fake news spreading [3]. To counter this threat, the forensic research community has started to develop a series of techniques that detect fake GAN-generated image [4]–[6].

All of the above-mentioned strategies are among the latest solutions for GAN image detection. However, the CGI detection problem has been extensively investigated in the past multimedia forensic literature [7]–[9]. It is worth noting that previous methods aimed at exposing some specific CGI inconsistencies and artifacts from some characteristic statistical traces or according to a pre-defined model. These strategies were suggested by the knowledge of the available CGI algorithms that could have been applied to generate the fake image. However, GAN-generated images can not be related to a well defined model, since each scheme presents its own peculiarities depending on the implemented architecture and training process. Indeed, as shown in [5], each architecture may introduce different traces, thus making their generalization a complex task. A detector that has been trained to detect images generated by a specific GAN architecture could not be suitable for a different GAN scheme.

For this reason, the approach analyzed in this paper focuses on identifying and analyzing statistical traces that make GAN-generated images differ from natural photographs. Previous work has shown that, on natural digital images, the probability distribution of specific variables usually follows a pre-defined behavior that proves to be completely-altered whenever the image is modified. As an example, the distribution of the first significant digit of quantized DCT coefficients follow Benford's law [10]. This property can be proved whenever the statistics of quantized DCT coefficients shows an exponential decay and can be empirically-verified on real images.

Exploiting this property, many forensics detectors have been successfully proposed in the literature (e.g., for detection of JPEG compression [11], [12], face morphing [13], [14] synthetic imagery, etc.). Despite these premises, there is no current proof that GAN-generated pictures should be statistically compliant to Benford's law [15].

In this work, we investigates whether Benford's law can be used for the detection of GAN-generated images. The reported analysis is exploited to design a GAN image detector which proved to be extremely accurate with a limited computational effort. More precisely, we verify that Benford's law is not followed by GAN images, and we propose a set of related features that could highlight this unfitting for an analyzed digital image. A simple supervised learning framework is then proposed to detect if an image is natural or GAN-generated from the extracted features.

This solution is evaluated on an image corpus made available by the authors of [16], enriched by additional images obtained by more modern GANs. We make use of more than 200 000 GAN-generated images obtained through different architectures trained on different tasks on different datasets. Results show that there is a trade-off between the chosen size of the proposed feature vector and the achieved accuracy. It is possible to either use a compact feature vector to obtain results comparable with the state-of-the-art, or a larger feature vector that allows improving against the more recent solutions proposed in the literature. This flexibility makes the proposed solution particularly suitable also for low-power devices not equipped with an advanced Graphics Processing Unit (GPU), which might still need to detect whether images are fake or not (e.g., smartphones, tablets, etc.). Additionally, we discuss resilience to JPEG compression in order to better define the working conditions of the proposed method.

II. Background

Benford's law, which is also known as First Digit (First Digit (FD)) law or Significant Digit law, concerns the statistical frequencies of the most significant digits for the elements of a real-life numerical set. More precisely, the rule states that, given a set of measurements for some natural quantities (e.g., population of cities, stock prices, etc.), the statistics of their FD follows the distribution depicted in Fig. 1 and described by the equation

$$p(d) = \log_{10}\left(1 + \frac{1}{d}\right), \tag{1}$$

where d is the FD in base 10 (the generalized version of this law is presented in the next section). This has been empirically-observed over a vast range of natural quantities [17], but it is also possible to prove it in closed form for many exponentially-decreasing probability distributions [18]. It has also been observed that this rule is not well-fitted by FD statistics from altered data: whenever numbers are changed according to some selective strategies, FD frequencies deviate from their theoretical values [19]. As a consequence, this proof has been used as supporting evidence for detecting falsified accounts, fake financial reports, and frauds [20].

This property has been largely exploited in multimedia forensics to detect image tampering. In fact, natural image DCT coefficients can be typically modeled by a Laplacian-like distribution [21], which naturally follow Benford's law, and for this reason, the mentioned rule can be successfully used in image forensic applications [22].

A well-known application of Benford's law in forensics is the study of JPEG compression traces [23]: the authors propose using such rule to verify if an image has been JPEG compressed once or twice. Milani et al. [12] exploit FD's features to detect multiple JPEG compression, also showing robustness against rotation and scaling. Pasquini et al. [11] address the multiple JPEG compression detection problem by means of Benford-Fourier analysis. The same authors also investigate traces of previous hidden JPEG compression in uncompressed images [24].

This rule has also been successfully applied to other forensic problems. In [25], the authors show that it is possible to leverage FD distribution to roughly estimate the amount of processing that has been applied to a given image. The authors of [26] apply Benford's law to solve image contrast enhancement detection problem. In [27], the authors make use of this law to deal with splicing forgery localization.

Another interesting application of Benford's law in image forensics is detecting computer graphics and computer generated images. To this purpose, Del Acebo et al. [14] model light intensity in natural and synthetic images, concluding that FD's law is not followed by the latter. Makrushin et al. [13] show how to efficiently detect morphed faces using the fitting parameters of the Benford's logarithmic curve as a features.

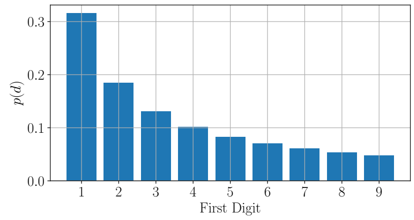


Fig. 1. Benford's law FD pmf considering base 10 for FD computation.

Anyway, detecting synthetic images is nowadays a timely and crucial forensic need due to the achievements of GAN technology in generating highly-realistic fake photographs. This possibility has been recently used to create false image and video contents in deepfake political propaganda, revenge porn, fake news creation. For these reasons, during the last years multimedia forensics researchers have been focusing on designing reliable strategies to detect synthetic images.

To this purpose, [4] proposes a method to detect image-to-image translation over social networks. Specifically, the authors compare different detectors fine-tuned for the binary classification task of GAN-generated against natural image detection. The same authors also show how a model-specific fingerprint can be retrieved by GAN generated images in order to identify the specific network used for image generation [5]. In [16], authors apply an incremental learning strategy to train a GAN-generated image detector that can be progressively updated in time as new images from different kinds of GANs are processed. In [28], the authors propose a method to detect GAN-generated images by analyzing the disparities in color components between real scene images and generated images. In [6], GAN images are detected by analyzing saturation artifacts in pixel distributions. Moreover, if videos are analyzed, methods exploiting also the temporal evolution of frames have been proposed [27], [29].

III. Motivations

Natural images, as many other natural processes, can be roughly approximated as autoregressive signals [30]. This is the rationale behind different historical as well as more recently proposed image compression [30], [31] and generation [32], [33] methods. From these assumptions, an image can be modeled as a complex autoregressive signal with a generally low-pass characteristics.

GAN generator's structures are usually composed by a concatenation of limited-support convolutional layers followed by non-linearities. Filters' coefficients are optimized so that GAN's response to a given input belongs to the desired output class. However, in most GAN implementations, practical and complexity reasons have led to the adoption of filters with a limited size. Therefore, if no recursive operations are applied in the network architecture, the output of a GAN generator looks more like a signal filtered through a Finite Impulse Response (FIR) filter than a complex autoregressive process.

The rationale behind the proposed method is that the information related to the filter ideally used to generate the data under analysis can be used to discriminate natural images (with autoregressive and complex spectra) from GAN-generated ones (generated through operations closer to FIR filtering). This can be done analyzing the statistics of quantized DCT coefficients.

More precisely, let us assume that an input grayscale image is partitioned into K distinct 8×8 blocks, which are then mapped into the 2D-DCT domain and further quantized. This processing chain is used by the JPEG coding standards and proves to be tailored to the spectral characteristics of images. Some of the past works highlight that, in the frequency domain, the quantized DCT coefficient statistics of natural images must follow Benford's law [22].

Let us denote as $c_{n,\Delta}(k)$ the DCT coefficient at the n-th frequency in zig-zag mode obtained from the k-th block and quantized with step Δ. It is possible to compute the corresponding FD with base b as

$$d_{b,n,\Delta}(k) = \left\lfloor \frac{|c_{n,\Delta}(k)|}{b^{\lfloor \log_b |c_{n,\Delta}(k)| \rfloor}} \right\rfloor. \tag{2}$$

As $d_{b,n,\Delta}(k)$ can only assume $b-1$ values (i.e., all possible digits defined in base b apart from zero), its pmf $\hat{p}_{b,n,\Delta}$ computed over the K blocks is composed by $b-1$ elements. For the sake of notational compactness, let us momentarily drop the indexes n, b, and Δ. We can formally define the pmf $\hat{p}(d)$ as

$$\hat{p}(d) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}_x(d(k)), \; d \in \{1, 2, \dots, b-1\}, \tag{3}$$

where

$$\mathbf{1}_x(y) = \begin{cases} 1 & \text{if } y = x, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

This pmf for a natural image must follow the generalized Benford's law equation

$$p(d) = \beta \log_b \left(1 + \frac{1}{\gamma + d^\delta} \right), \tag{5}$$

where β is a scale factor, γ and δ parameterize the logarithmic curve, and $d \in \{1, 2, ..., b-1\}$ is one possible value of the considered first digits in base b.

The fitness between $\hat{p}(d)$ and $p(d)$ can be measured by some divergence functions such as the Jensen-Shannon divergence $D^{\text{JS}}(\hat{p}|p)$

$$D^{\text{JS}}(\hat{p}|p) = D^{\text{KL}}(\hat{p}|p) + D^{\text{KL}}(p|\hat{p}), \tag{6}$$

which is a symmetrized version of the well-known Kullback-Leibler divergence

$$D^{\text{KL}}(\hat{p}|p) = \sum_{d=1}^{b-1} \hat{p}(d) \log \frac{\hat{p}(d)}{p(d)}. \tag{7}$$

Since D^{JS} proves to be unstable for biased pmfs, it is possible to use the symmetrized Renyi divergence

$$D_\alpha^{\text{R}}(\hat{p}|p) = \frac{1}{1-\alpha} \left(\log S_\alpha(\hat{p}, p) + \log S_\alpha(p, \hat{p}) \right), \tag{8}$$

or the symmetrized Tsallis divergence

$$D_\alpha^{\text{T}}(\hat{p}|p) = \frac{1}{1-\alpha} \left(2 - S_\alpha(\hat{p}, p) - S_\alpha(p, \hat{p}) \right), \tag{9}$$

where

$$S_\alpha(q, p) = \sum_{d=1}^{b-1} q(d)^\alpha / p(d)^{\alpha - 1}. \tag{10}$$

It is possible to prove that, whenever an image is altered (e.g., it is compressed/quantized a second time, etc.), Benford's law is not verified anymore. In fact, many modifications redistribute image coefficients among the bins of the quantizer, thus the final pmf associated to quantized DCT coefficients presents some oscillating probability values that deviate from the ideal distribution. For these reasons, many solutions in the past measured the divergence between the empirically-estimated $\hat{p}(d)$ and its ideal fitted version $p(d)$ in order to find whether the image has been altered or not. In this paper we show that it is possible to adopt the same solution to detect GAN-generated pictures.

IV. GAN-GENERATED IMAGE DETECTION

In this section we provide a formal definition of the GAN-generated image detection problem and report all the technical details about the detection method we propose.

A. Problem formulation

We define the GAN-generated image detection problem as a two-class classification problem. Given an image I, we want to understand whether it has been synthetically generated through a GAN, or it is a natural photograph.

Formally, to solve this problem we consider a pipeline composed by two blocks: a feature extractor and a supervised classifier. The feature extractor implements the function $\Phi(\cdot)$, which turns the image into a more compact yet informative representation, i.e., the feature vector $\boldsymbol{\phi} = \Phi(I)$. The classifier implements the function $M(\cdot)$ such that: $M(\boldsymbol{\phi}) = 0$ if the image is a natural one; $M(\boldsymbol{\phi}) = 1$ if the image comes from a GAN. With this framework in mind, we focus on designing the function $\Phi(\cdot)$ based on Benford's law, so that a simple classifier can be effectively used.

B. Detection method

The feature extraction process is depicted in Fig. 2. Given an image I, we divide it in K non-overlapping blocks with resolution 8×8 pixel. From each block, we compute its 2D-DCT representation. We then quantize it using a given quantization step Δ (chosen for each coefficient according to a JPEG quantization matrix).

Given a base b, we compute the first digit of the n-th quantized 2D-DCT frequency sample from the k-th block according to (2). We then compute the pmf $\hat{p}_{b,n,\Delta}$ according to (3). Examples of $\hat{p}$ for different bases for both natural and GAN-generated images are reported in Fig. 3. Finally, we fit generalized Benford's law expressed in (5) by solving a mean square error minimization problem as

$$p_{b,n,\Delta}^{\text{fit}} = \arg\min_p \sum_{d=1}^{b-1} (\hat{p}_{b,n,\Delta}(d) - p(d))^2. \tag{11}$$

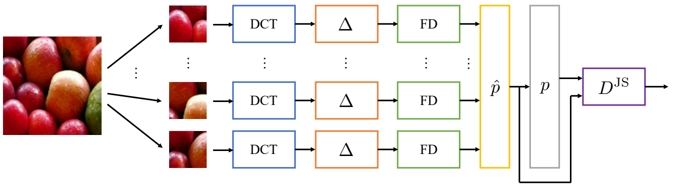
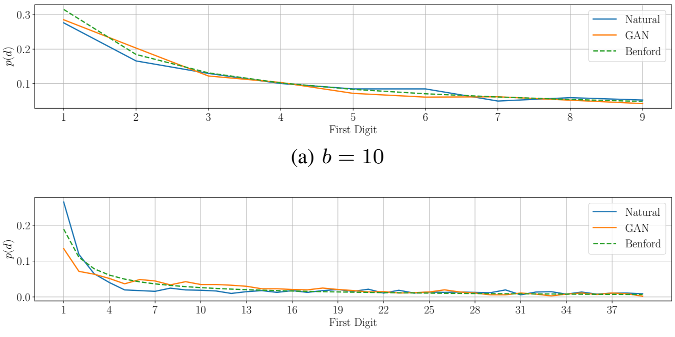


Fig. 2. Feature extraction pipeline considering a single divergence, quantization step Δ, base b and DCT coefficient n. Extraction process is repeated for multiple parameters.



(a) $b = 10$

(b) $b = 40$

Fig. 3. Different pmf $\hat{p}$ for natural (blue) and GAN-generated images (oranges) are compared to the ideal Benford curve (dashed green) for different bases b. Blue and orange curves deviates differently from the green one.

Comparing the computed pmf $\hat{p}_{b,n,\Delta}$ and the Benford fit $p^{\text{fit}}_{b,n,\Delta}$, we compute Jensen-Shannon divergence $D^{\text{JS}}_{b,n,\Delta}$, Renyi divergence $D^{\text{R}}_{b,n,\Delta}$, and Tsallis divergence $D^{\text{T}}_{b,n,\Delta}$ as reported in Section III. Notice that we removed the dependency of Tsallis and Renyi divergence on α as we keep it constant in our experiments.

Finally, considering a set $\mathcal{B}$ of bases, a set $\mathcal{N}$ of DCT frequencies and a set $\mathcal{J}$ of JPEG quality factors driving the quantization parameter Δ, we obtain the final feature vector by concatenating all divergences as:

$$\phi_{\mathcal{B},\mathcal{N},\mathcal{J}} = [D^{\text{JS}}_{b,n,\Delta},\ D^{\text{R}}_{b,n,\Delta}\ D^{\text{T}}_{b,n,\Delta}]_{b\in\mathcal{B},n\in\mathcal{N},\Delta\in\mathcal{J}}. \tag{12}$$

Notice the the feature vector size depends on how many DCT coefficients, bases and quantization steps are used during the analysis. For instance, if we choose a single compression step, a single DCT frequency and a single base, the feature vector will be composed by the concatenation of just three divergences, thus having dimensionality 3. Conversely, if we use multiple bases, frequencies and compression steps, we end up with a bigger vector. In our experiments, we consider vectors with dimensionality ranging from 3 to 540, as shall be explained in Section V.

After feature computation, the vector $\phi_{\mathcal{B},\mathcal{N},\mathcal{J}}$ is fed to a supervised classifier. In order to study the effectiveness of Benford-based features, we do not adopt unnecessarily complicated classifiers $M(\cdot)$. Specifically, we resort to a Random Forest classifier.

V. Results

In this section we discuss the used datasets, the experimental setup, and finally report all the results achieved with the proposed technique for GAN-generated image detection.

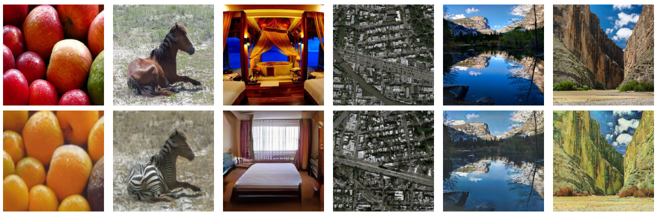


Fig. 4. Examples of original (top) and GAN-generated (bottom) images belonging to the dataset proposed in [16].

TABLE I
DATASET COMPOSITION

Architecture	Dataset	Number of images
Cycle-Gan	orange2apple	1280
	photo2ukiyoe	4072
	winter2summer	1484
	zebra2horse	1670
	photo2cezanne	3978
	photo2vangogh	4099
	photo2monet	4765
	facades	259
	cityscapes	1996
	sats	684
ProGAN	lsun_bedroom	30770
	lsun_bridge	28768
	lsun_churchoutdoor	29120
	lsun_kitchen	42706
	lsun_tower	29020

A. Dataset

In order to build our dataset, we started from the publicly available GAN-dataset released by Marra et al. [16]. Specifically, we considered a corpus composed by 15 different sub-datasets of images obtained employing 2 different architectures: Cycle-Gan [34] and ProGAN [35]. The first architecture is designed for image-to-image translation purpose, i.e., mapping an image of a given class (i.e., pictures of horses) to an image of another one (i.e., pictures of zebras). The second architecture is a generator able of creating natural looking pictures of different scenes depending on the used training data (e.g., bedroom pictures, bridges, etc.). Each dataset comprises both natural images and their GAN-generated counterparts. All images are color images and have a resolution of 256 × 256 pixel. The complete composition is reported in Table I and some examples of the more than 200 000 images are reported in Fig. 4.

B. Setup

As shown in Section IV, each feature depends on a selected set of bases $\mathcal{B}$, DCT frequencies $\mathcal{N}$, and analysis JPEG Quality Factor (QF) describing the set of quantization steps $\mathcal{J}$. With regards to bases, we test all combinations of sets of bases $\mathcal{B} \subseteq \{10, 20, 40, 60\}$ containing from one to four elements. This leads to 15 possible combinations of bases (i.e., from $\mathcal{B} = \{10\}$ to $\mathcal{B} = \{10, 20, 40, 60\}$). Concerning

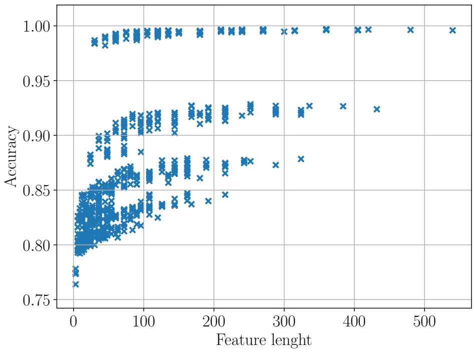


Fig. 5. Accuracy obtained with different feature vectors obtained changing the considered sets $\mathcal{B}$, $\mathcal{N}$ and $\mathcal{J}$. Each vector has a different length and provides a different accuracy result.

the selected DCT frequencies, we choose a limited amount of sets $\mathcal{N} \subseteq \{1, 2, \dots, 9\}$ (i.e., including the first 9 frequencies in zig-zag order after DC) , similarly to other detectors available in literature [12]). Specifically, we only consider 9 sets obtained by progressively-adding one frequency at a time from the previous set (i.e., $\mathcal{N} = \{1\}$, $\mathcal{N} = \{1, 2\}$, ..., $\mathcal{N} = \{1, 2, \dots, 9\}$). As for the quantization step values, the final feature set was created concatenating the 5 arrays of divergences obtained using JPEG QFs in the set $\mathcal{J} \subseteq \{80, 85, 90, 95, 100\}$. Considering all combinations of bases, frequencies and JPEG quantization steps, we obtain a total amount of 675 different setups.

For each feature vector described in Section IV (i.e., each setup), we trained a different Random Forest classifier performing Leave-One-Group-Out cross-validation over the various datasets as explained in [4]. The choice of Random Forest was suggested by the low complexity requirements, the generalization capabilities, and the resilience to small training datasets. Namely, given a dataset $\mathcal{D}_i$ out of the complete set of dataset $\mathcal{D}$, we trained our model over the remaining $\mathcal{D}_j$, $\forall j \neq i$ and we test over $\mathcal{D}_i$. Results are always shown on the leave-out dataset, and we report the maximum accuracy value among all the different setups. To provide a practical example, let us consider the situation in which we test on dataset $\mathcal{D}_i = \texttt{orange2apple}$. This consists of original images (i.e., apples and oranges) and GAN images (apples turned into oranges and viceversa). The classifier was trained on all the other images (excluding those in orange2apple) in order to avoid biasing the results with overfitting. We adopted the Random Forest implementation provided with the open source Scikit Learn Python library. After a grid search over several candidates, we fixed the number of Decision Trees to 100, with bootstrap sampling enabled. We selected Gini index as splitting policy, leaving all the other parameters as their default values.

C. Experiments

In this section we report all experimental results achieved to evaluate the proposed technique. Moreover, we report a comparison against baseline solutions. Finally, we provide some additional insights in terms of resilience to JPEG compression.

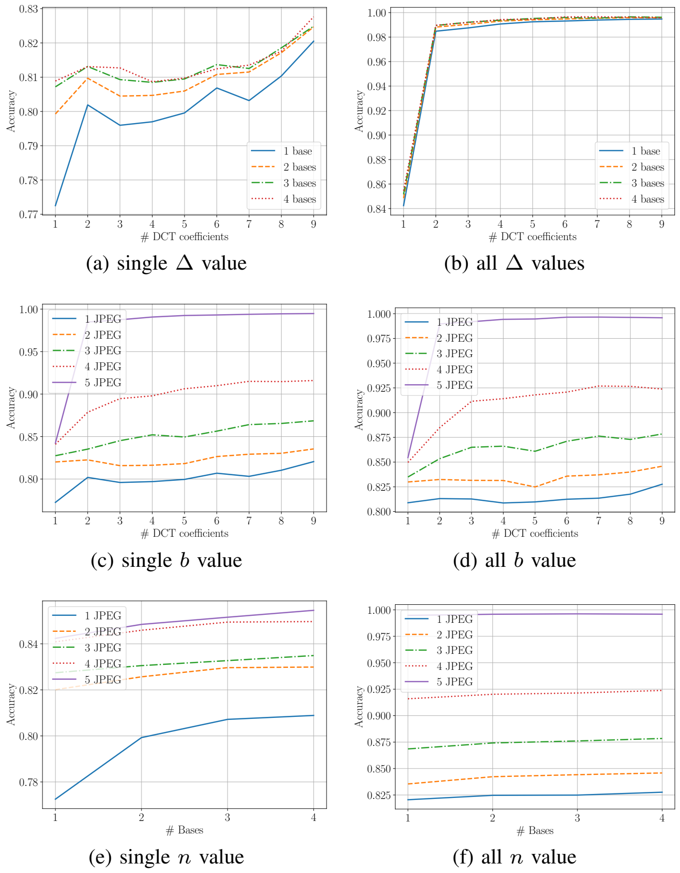
To select the baselines, we focused on the work proposed by Marra et al. [4] since, to the best of our knowledge, it is the only work to perform an extensive GAN detection test over a large dataset of images. Specifically, we selected two baselines: a completely data-driven one based on deep learning; a solution based on hand-crafted features commonly used in the forensics literature.

Similarly to the solution in [4], we compared our approach with the Xception Convolutional Neural Network (CNN), as the first baseline method. According to the results in [4], this set of features seems to provide the best results over most of the considered datasets. Starting from the pretrained model, we finetuned it on our dataset, following the same Leave-One-Group-Out strategy we adopted for the Random Forest training. We used 70% of the training data for the actual training, and the remaining 30% for validation, testing on the Leave-Out dataset. We resorted to Adam optimization algorithm, with an initial learning rate of 0.0001, training until reaching convergence on a validation plateau. We adopted the Keras implementation of Xception, performing the training in several hours on a workstation equipped with a NVIDIA Titan V GPU, a Intel Xeon E5-2687W and 256 GB of RAM.

The second baseline method operates a linear Support Vector Machine (SVM) on a set of handcrafted steganalysis rich-features (as suggested in [4]). These features have been successfully used in image forgery detection tasks as well [36]. The model has been trained following the same train/test strategy used for Xception, using the Scikit Learn implementation of Support Vector Machine (SVM).

Feature length and parameters. In the design of the proposed solution we considered different combinations of features obtained varying the parameters in the set $\mathcal{B}$, $\mathcal{N}$, $\mathcal{I}$ and changing feature vectors' lengths. As a matter of fact, it is necessary to evaluate how the vector length could impact on the classifier performance. Fig. 5 shows the average test accuracy obtained on all datasets considering all possible 675 feature vectors. It is possible to notice that even the smallest feature vectors of just 3 elements enable achieving an accuracy greater than 0.75. It is sufficient to use 50 features to have accuracy higher than 0.97.

In order to gain a better insight on the effect of using different bases, DCT coefficients and quantization steps, we performed an analysis by keeping some parameters fixed, and just changing the others. Fig. 6(a) and (b) show the results with a single fixed quantization step value, and considering all the values, respectively. In both scenarios it is possible to notice that the greatest improvement is obtained when more than a single DCT coefficient is used. Moreover, the more the coefficients, the better the results. Fig. 6(c) and (d) report the results with features from a single fixed FD base and from all the considered bases, respectively. It seems that using more than one base only marginally improve the results. As a matter of fact, both figures are very similar. Finally, Fig. 6(e) and (f) display the accuracy values obtained from the features of a single DCT frequency and of the whole set of frequencies, respectively. . From these figures it is possible to note that



(a) single Δ value

(b) all Δ values

(c) single b value

(d) all b value

(e) single n value

(f) all n value

Fig. 6. Accuracy varying different parameters: (a) fix 1 JPEG compressions; (b) fix 5 JPEG compressions; (c) fix 1 base; (d) fix 4 bases; (e) fix 1 DCT coefficient; (f) fix 9 DCT coefficients. When we fix a single parameter, we average all results obtained by fixing that parameter to each possible value it can span.

the more the considered quantization steps, the higher the accuracy for any other parameters set. This is not particularly surprising, as Benford's law is naturally linked to the used JPEG quantization.

Comparison against baseline. In order to compare against the selected baselines solution, we finetuned Xception network and trained a linear SVM on the steganalysis features for each dataset according to the same procedure used for our Random Forest as suggested in [4]. Table II reports the breakdown of test accuracy scores for all datasets. The highest average accuracy among the considered methods is obtained by the proposed method, and it is higher than 0.99. It is also interesting to notice that the proposed solution is considerably better than the baseline CNN on `winter2summer`, `sats` and `lsun_bedroom`, which seem to be particularly though for the latter. These results highlight that, in order to properly train a very deep network like Xception, a much larger dataset probably is needed. However, this might be difficult to obtain in a reduced amount of time in a forensic scenario. On the contrary, the proposed feature vector is very compact, thus Random Forest does not suffer from a smaller training set.

TABLE II

ACCURACY RESULTS COMPARED TO THE BASELINE SOLUTIONS FOR EACH DATASET. AVERAGE ACCURACY (AVG) IS ALSO REPORTED. BEST RESULT PER DATASET IN BOLD.

Dataset	Proposed	Xception	Steganalysis
orange2apple	**98.13**	97.64	88.80
photo2ukiyoe	**100.00**	97.41	86.78
winter2summer	**100.00**	68.33	77.96
zebra2horse	**99.69**	89.58	91.01
photo2cezanne	**99.97**	95.91	95.88
photo2vangogh	**100.00**	93.75	94.68
photo2monet	**99.84**	94.08	94.80
facades	**100.00**	99.84	73.93
cityscapes	**100.00**	**100.00**	**100.00**
sats	**99.69**	73.00	90.92
lsun_bedroom	**100.00**	76.22	98.92
lsun_bridge	**99.89**	82.49	95.90
lsun_churchoutdoor	**99.99**	99.79	98.81
lsun_kitchen	**99.99**	87.26	99.49
lsun_tower	**99.98**	95.45	98.87
avg	**99.83**	89.64	91.03

The baseline handcrafted method performs reasonably well, but the obtained accuracy is lower than that of the proposed method of almost 9% on average.

Resilience to JPEG compression. When images are shared online, JPEG compression is almost always applied in order to reduce network and storage requirements. Therefore, we measured the performance of the proposed method whenever a further JPEG compression is applied with different coding parameter configurations.

In a first scenario, GAN-generated and real images have been randomly JPEG compressed considering quality factors distributed in $\{85, \dots, 100\}$. The originally-trained detector (on non-compressed images) was then tested on this newly compressed dataset In this situation, the proposed solution approaches a random guess accuracy. However, this situation is not completely unexpected. As a matter of fact, Benford's law is strictly tailored to JPEG compression. Therefore, scrambling with JPEG coefficients statistics through recompression has a high impact on Benford's features.

We therefore considered a second scenario, which is more realistic as shown in [4]. If we know that images might be JPEG recompressed, we can also train our system on JPEG compressed images. We therefore re-trained our method and the baseline on compressed images, and tested them on compressed images. In this situation, results improve as expected. As a matter of fact, the proposed solution accuracy decreases, but still remains higher than 0.80. In particular, results depend on the specific datasets and GAN architecture. Indeed, all results related to ProGAN (i.e., last five datasets) show an almost optimal accuracy always higher than 0.99. Conversely, on Cycle-Gan images, only a couple of datasets exhibit accuracy grater than 0.70. In this situation, if computational is feasible for the adopted architecture, the baseline network might be preferable.

We then tested a third scenario, assuming that the analyst knows which is the quality factor adopted by the final JPEG

TABLE III
ACCURACY OBTAINED USING DIFFERENT JPEG QUALITY FACTORS.

QF	Dataset	Proposed	Xception
100	orange2apple	**94.50**	92.56
	photo2ukiyoe	**100.00**	98.50
	cityscapes	**100.00**	100.00
	lsun_tower	**100.00**	94.64
95	orange2apple	82.01	**90.66**
	photo2ukiyoe	97.00	**98.42**
	cityscapes	**99.99**	99.32
	lsun_tower	**99.80**	99.48
90	orange2apple	65.93	**85.61**
	photo2ukiyoe	92.01	**98.17**
	cityscapes	**100.00**	99.66
	lsun_tower	99.60	**98.86**

compression stage (since it can be read from the bit stream). It is possible to train a different Random Forest classifier or Xception network for each quality factor. We therefore generated three versions of the dataset by recompressing it with quality factor 100, 95, and 90, respectively. For each quality factor, we trained the proposed method and Xception baseline using the aforementioned leave-one-group-out strategy. We did not consider steganalysis features anymore, as in [4] the authors already showed that they greatly suffer JPEG compression. Results are reported in Table III. It is possible to notice that for high quality factors, the proposed Benford-based method outperforms the baseline. Xception network shows better results starting from quality factor 90.

In the final testing scenario, we assume that the analyst wants to train a different classifier for each JPEG quality factor, and for each kind of image content. As an example, if the analyst is interested in detecting fake oranges with a given quality factors, he/she might train only on the orange2apple dataset, rather than the others. In this situation (i.e., known quality factor and kind of GAN training dataset), both the proposed method and the Xception baseline achieve an almost perfect result for each quality factor (i.e., 100, 95 and 90).

D. Analysis on faces.

All results shown so far are obtained not considering GANs generating face images. This is due to two main reasons. First, GANs that were trained to generate face images produce particularly realistic results lately. This make face images harder to detect as GAN-generated compared to other kind of imagery. Indeed, shadows and lightning very often respect physics law, thus making Benford's law almost verified [14]. Second, face-generating GANs are often trained on common pristine face datasets [37], which makes the Leave-One-Group-Out testing strategy applied to now impracticable.

In the light of these considerations, we decided to create a specific dataset, composed by all the faces dataset in the original corpus, generated by ProGAN [35] (19 870 images), StarGAN [38] (50 000 images) and GlowGAN [39] (49 900 images), plus some additional images generated by the more recently proposed StyleGAN2 [40] (2000 images). As pristine faces, we always consider images from the Celeb-A dataset [37].

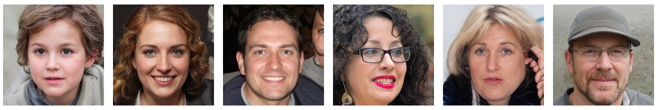


Fig. 7. Examples of faces generated with StyleGan2 [40].

TABLE IV
ACCURACY RESULTS FOR EACH FACE TEST DATASET. AVERAGE ACCURACY (AVG) IS ALSO REPORTED. ACCURACY HIGHER THAN 85% ARE REPORTED IN BOLD.

Dataset	Proposed
progan_celeba	79.75
stargan_black_hair	97.26
stargan_blond_hair	96.56
stargan_brown_hair	96.76
stargan_male	96.24
stargan_smiling	96.06
glow_black_hair	86.56
glow_blond_hair	88.26
glow_brown_hair	86.18
glow_male	87.11
glow_smiling	83.04
stylegan2-0.5	77.18
stylegan2-1	72.63
avg	87.96

ProGAN is trained to generate realistic faces similar to those from Celeb-A dataset [37]. StarGAN and GlowGAN are trained to obtain faces with different characteristics (e.g., hair colors, smiles, etc.). Finally, StyleGAN2 produces images at different qualities depending on its configuration parameter $\psi = 0.5$ or $\psi = 1$ as suggested by the authors [40], starting from the Flickr-Faces-HQ Dataset [35]. Some random images from those generated by StyleGAN2 are shown in Fig. 7. For each dataset, we train a Random Forest classifier considering 70% of the images as training set and 30% as test.

Table IV shows the achieved results on each test set. It is possible to notice that StarGAN and GlowGAN seems to be easier to detect. On the contrary, ProGAN and StyleGAN2 looks more challenging. These promising preliminary results motivate some future work with more extended face image datasets, also comparing against other baselines and in presence of editing operations.

VI. CONCLUSIONS

In this paper we proposed a study on the use of the well-known Benford's law for the task of GAN-generated image detection. We proposed a strategy to extract Benford-related features from an image relying on different divergence definitions. We also showed how to combine these features in order to better exploit different bases as well as DCT frequencies. Using these features, we performed a series of experiment based on a simple Random Forest classifier in order to study the amount of information captured by the features, rather than focusing on specializing a complex classifier.

Results show that GAN-generated images often fail in respecting Benford's law, thus can be discriminated from natural pictures. However, some kind of CNN architectures seem to produce images that are harder to detect than others. This motivates future studies on this topic. Moreover, this suggests to possibly embed Benford's law into GAN loss function in order to obtain even more realistic images.

References

[1] A. Bansal, S. Ma, D. Ramanan, and Y. Sheikh, "Recycle-GAN: Unsupervised video retargeting," in *European Conference on Computer Vision (ECCV)*, 2018.

[2] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] M. Brundage *et al.*, "The malicious use of artificial intelligence: Forecasting, prevention, and mitigation," *arXiv:1802.07228*, Feb. 2018.

[4] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva, "Detection of GAN-Generated Fake Images over Social Networks," *IEEE International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018.

[5] F. Marra, D. Gragnaniello, L. Verdoliva, and G. Poggi, "Do GANs Leave Artificial Fingerprints?" *IEEE International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2019.

[6] S. McCloskey and M. Albright, "Detecting GAN-generated imagery using saturation cues," in *IEEE International Conference on Image Processing (ICIP)*, 2019.

[7] H. Farid and M. J. Bravo, "Perceptual discrimination of computer generated and photographic faces," *Digital Investigation*, vol. 8, pp. 226–235, 2012.

[8] D. Dang-Nguyen, G. Boato, and F. G. B. De Natale, "3d-model-based video analysis for computer generated faces identification," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 10, pp. 1752–1763, 2015.

[9] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *IEEE Workshop on Information Forensics and Security (WIFS)*, 2017.

[10] F. Pérez-González, G. L. Heileman, and C. T. Abdallah, "Benford's law in image processing," in *IEEE International Conference on Image Processing (ICIP)*, 2007.

[11] C. Pasquini, G. Boato, and F. Pérez-González, "Multiple JPEG compression detection by means of Benford-Fourier coefficients," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014.

[12] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple JPEG compressions using first digit features," *APSIPA Transactions on Signal and Information Processing*, vol. 3, pp. 1–11, 2014.

[13] A. Makrushin, C. Kraetzer, T. Neubert, and J. Dittmann, "Generalized Benford's law for blind detection of morphed face images," in *ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, 2018.

[14] E. del Acebo and M. Sbert, "Benford's law for natural and synthetic images." in *Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, 2005.

[15] F. Benford, "The law of anomalous numbers," *Proceedings of the American philosophical society*, 1938.

[16] F. Marra, C. Saltori, G. Boato, and L. Verdoliva, "Incremental learning for the detection and classification of GAN-generated images," *arXiv:1910.01568v2*, 2019.

[17] R. A. Raimi, "The first digit problem," *The American Mathematical Monthly*, vol. 83, no. 7, pp. 521–538, 1976.

[18] J. Wang, B. Cha, S. Cho, and C. . J. Kuo, "Understanding benford's law and its vulnerability in image forensics," in *2009 IEEE International Conference on Multimedia and Expo*, June 2009, pp. 1568–1571.

[19] A. Diekmann, "Not the first digit! using Benford's law to detect fraudulent scientific data," *Journal of Applied Statistics*, vol. 34, pp. 321–329, 04 2007.

[20] K.-H. Todter, "Benford's law as an indicator of fraud in economics," *German Economic Review*, vol. 10, pp. 339–351, 08 2009.

[21] S. Smoot and L. Rowe, "Study of DCT coefficient distributions," in *SPIE Symposium on Electronic Imaging (EI)*, 1996.

[22] F. Pérez-González, T. T. Quach, C. Abdallah, G. L. Heileman, and S. J. Miller, "Application of benford's law to images," in *Benford's Law: Theory and Applications*, S. Miller, Ed. Princeton University Press, 2015, ch. 19.

[23] T. Pevny and J. Fridrich, "Detection of double-compression in jpeg images for applications in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 2, pp. 247–258, June 2008.

[24] C. Pasquini, G. Boato, and F. Pérez-González, "Statistical detection of JPEG traces in digital images in uncompressed formats," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 12, no. 12, pp. 2890–2905, Dec 2017.

[25] S. Milani, M. Fontana, P. Bestagini, and S. Tubaro, "Phylogenetic analysis of near-duplicate images using processing age metrics," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[26] S. S. Moin and S. Islam, "Benford's law for detecting contrast enhancement," in *International Conference on Image Information Processing (ICIIP)*, 2017.

[27] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, 2019.

[28] H. Li, B. Li, S. Tan, and J. Huang, "Detection of deep network generated images using disparities in color components," *arXiv preprint arXiv:1808.07276*, 2018.

[29] D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," *IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, 2019.

[30] E. J. Delp, R. L. Kashyap, and O. R. Mitcheli, "Image data compression using autoregressive time series models," *Pattern Recognition*, vol. 11, pp. 313–323, 1979.

[31] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 10 771–10 780.

[32] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," *CoRR*, vol. abs/1601.06759, 2016. [Online]. Available: http://arxiv.org/abs/1601.06759

[33] A. van den Oord, N. Kalchbrenner, L. Espeholt, k. kavukcuoglu, O. Vinyals, and A. Graves, "Conditional image generation with pixelcnn decoders," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4790–4798.

[34] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[35] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *International Conference on Learning Representations*, 2018.

[36] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery detection through residual-based local descriptors and block-matching," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 5297–5301.

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

[38] Y. Choi, M. Choi, M. Kim, J.-W. 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*, 2018.

[39] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *NeurIPS*, 2018.

[40] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," *CoRR*, vol. abs/1912.04958, 2019.