Mastering Deepfake Detection: A Cutting-edge Approach to Distinguish GAN and Diffusion-model Images

LUCA GUARNERA, Department of Mathematics and Computer Science, University of Catania, Catania, Italy
OLIVER GIUDICE, Banca D'Italia, Applied Research Team, IT dept., Rome, Italy
SEBASTIANO BATTIATO, Department of Mathematics and Computer Science, University of Catania, Catania, Italy

Detecting and recognizing deepfakes is a pressing issue in the digital age. In this study, we first collected a dataset of pristine images and fake ones properly generated by nine different Generative Adversarial Network (GAN) architectures and four Diffusion Models (DM). The dataset contained a total of 83,000 images, with equal distribution between the real and deepfake data. Then, to address different deepfake detection and recognition tasks, we proposed a hierarchical multi-level approach. At the first level, we classified real images from AI-generated ones. At the second level, we distinguished between images generated by GANs and DMs. At the third level (composed of two additional sub-levels), we recognized the specific GAN and DM architectures used to generate the synthetic data. Experimental results demonstrated that our approach achieved more than 97% classification accuracy, outperforming existing state-of-the-art methods. The models obtained in the different levels turn out to be robust to various attacks such as JPEG compression (with different quality factor values) and resize (and others), demonstrating that the framework can be used and applied in real-world contexts (such as the analysis of multimedia data shared in the various social platforms) for support even in forensic investigations to counter the illicit use of these powerful and modern generative models. We are able to identify the specific GAN and DM architecture used to generate the image, which is critical in tracking down the source of the deepfake. Our hierarchical multi-level approach to deepfake detection and recognition shows promising results in identifying deepfakes allowing focus on underlying task by improving (about 2% on the average) standard multiclass flat detection systems. The proposed method has the potential to enhance the performance of deepfake detection systems, aid in the fight against the spread of fake images, and safeguard the authenticity of digital media.

CCS Concepts: • **Computing methodologies → Supervised learning by classification**;

Additional Key Words and Phrases: Deepfake detection, generative adversarial nets, diffusion models, multimedia forensics

ACM Reference Format:
Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. 2024. Mastering Deepfake Detection: A Cutting-edge Approach to Distinguish GAN and Diffusion-model Images. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 11, Article 343 (September 2024), 24 pages. https://doi.org/10.1145/3652027

Authors' addresses: L. Guarnera (Corresponding author) and S. Battiato, Department of Mathematics and Computer Science, University of Catania, Viale Andrea Doria 6, Catania, Italy, Italy, 95126; e-mails: luca.guarnera@unict.it, sebastiano.battiato@unict.it; O. Giudice, Banca D'Italia, Applied Research Team, IT dept., Rome, Italy; e-mail: oliver.giudice@bancaditalia.it.

ACM 1551-6865/2024/09-ART343
https://doi.org/10.1145/3652027

1 INTRODUCTION

The term deepfake refers to all the multimedia contents generated by some specific AI model. **Generative Adversarial Networks (GANs)** [14] and **Diffusion Models (DMs)** [22, 41] are two distinct approaches in the field of generative modeling, with their own unique characteristics. The most common deepfake creation solutions are those based on GANs [14], which are effectively able to create from scratch or manipulate multimedia data. GANs are composed of two neural networks, a *Generator (G)* and a *Discriminator (D)*, that work concurrently through a process of competition. The generator engine creates synthetic data, while the discriminator evaluates the authenticity of both real and generated data. This adversarial process leads to the generator producing increasingly realistic content. Several surveys on methods dealing with GAN-based approaches for the creation and detection of deepfakes have been proposed in References [32, 34].

However, DMs are a class of probabilistic models that rely on simulating a diffusion process in reverse settings. DMs [22, 41] are arousing interest, thanks to their photo-realism and also to a wide choice in output control given to the user. DMs aims to model complex data distributions by iteratively adding noise to a random noise vector input for the generation of synthetic data. Stable Diffusion [36] and DALL-E 2 [35] are the most famous state-of-the-art DMs based on the text-to-image translation operation. While GANs have been known for their exceptional ability to generate high-quality images, DMs offer advantages in terms of better control over the generative process and the ability to generate images with a wide range of styles and content [9]. DMs are able to produce even better realistic images than GANs, since GANs generate high-quality samples but are demonstrated to fail in covering the entire training data distribution. Figure 1 shows graphically the typical computational frameworks of GAN and DM models.

To effectively counteract the illicit use of synthetic data generated by GANs and DMs, new deepfake detection and recognition algorithms are needed. As far as image deepfake detection methods in state-of-the-art are concerned, they mostly focus on binary detection (Real vs. AI generated [43, 46]). Interesting methods in state-of-the-art already demonstrated to effectively discriminate between different GAN architectures [13, 16, 44]. Methods to detect DMs and recognize them have been proposed just recently [8, 39].

To level up the deepfake detection and recognition task, we propose to classify an image among 14 different classes: 9 GAN architectures, 4 DMs engines, and 3 pristine datasets (labeled as belonging to the same "real class) by employing a novel multi-level hierarchical approach exploiting deep learning models properly trained and tested. The proposed approach consists of 3 levels of classification: (Level 1) Real vs. AI-generated images; (Level 2) GANs vs. DMs; (Level 3) recognition of specific AI (GAN/DM) architectures among those represented in the collected dataset. The best models individually trained to solve specific tasks at each level are based on ResNET-101 [20] (selected among others [11, 23, 42, 45] after extensive evaluation). Figure 2 shows the overall scheme of the proposed methodology. The overall framework has the potential to enhance the performance of deepfake detection systems, aid in the fight against the spread of fake images, and safeguard the authenticity of digital media. Experimental results demonstrated the effectiveness of the proposed solution, achieving more than 97% accuracy on average, exceeding the state-of-the-art. Moreover, the hierarchical approach can be used to analyze multimedia data in depth to reconstruct its history (forensic ballistics) [17], a task poorly addressed by the scientific community on synthetic data.

The main contributions of this research are the following:

- We propose a new Deepfake detector based on a hierarchical approach with three levels composed of different ResNet-101 models, individually trained to solve specific tasks. In particular:
  - Level 1 defines whether a multimedia content is Real or Deepfake;

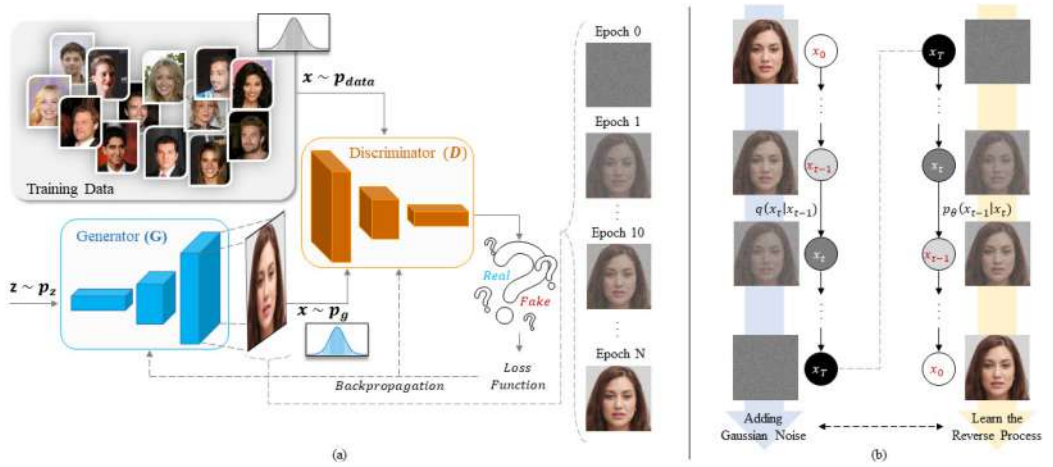


Fig. 1. (a) Generic GAN learning structure consisting of a Generator (G) that creates data samples from random input noise with the goal of learning the data distribution of the training set and generating new items (or manipulating them) of the same semantics; a Discriminator (D) attempts to distinguish between real data and generated by G. G and D are trained together in an adversarial manner. (b) Learning framework of a generic diffusion model: A latent variable model maps the latent space using a fixed Markov chain, gradually adding Gaussian noise to the data to obtain the approximate posterior $q(x_t|x_{t-1})$. The goal of training is to learn the reverse process ($p_\theta(x_{t-1}|x_t)$) so by running backward through this chain, new data can be generated. $x_1, \ldots, x_T$ are the latent variables with the same dimensionality as x_0.

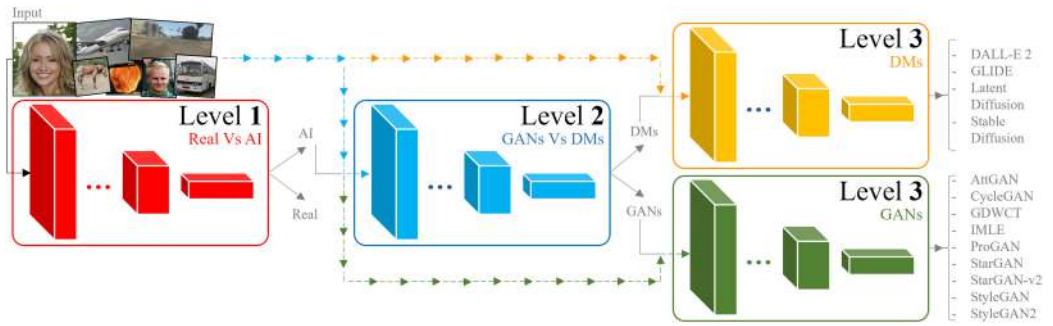


Fig. 2. Execution flow of the proposed hierarchical approach. Level 1 classifies images as real or AI-generated. Level 2 defines whether the input images were created by GAN or DM technologies. Level 3, composed of two sub-modules, solves the AI architecture recognition task. The dashed arrows represent an optional flow (e.g., in the case the input image is real, it will not be analyzed by the next levels).

- Level 2 defines the specific framework used for the creation between GAN and Diffusion Models;
- Level 3, divided into two sub-levels (one specialized on DM engines—Level 3-DM—and one on GAN architectures—Level 3-GAN) addresses the architecture recognition task on 13 different generative models (4 DM and 9 GAN).

— The experimental results demonstrate the potentials of the proposed method not only in solving the classical Real vs. Deepfake binary classification with an accuracy of 98.93% at level 1, but also in solving specific tasks: accuracy of 98.45% at level 2; 99.37% at level 3-DM; 97.01% at level 3-GAN. Overall, the entire framework, considering the accumulation

of classification error across levels, achieves a classification accuracy of 97.82%, which exceeds the 95.71% of a classic ResNET-101 "flat" model, i.e., trained on all 14 classes considered in this work.
— Compared with the various approaches in the literature, the hierarchical method succeeds in analyzing an image in detail, trying to reconstruct its history [3] with high precision. This turns out to be very important in forensics, as we begin to introduce the concept of explainability, step-by-step, with Deep Learning techniques.
— The models of the hierarchical approach turn out to be robust to various attacks on digital images such as Gaussian blur, mirroring, scaling, rotation, JPEG compression.
— The proposed method achieves excellent generalization results on datasets and contexts never seen before: COCOfake [2] (dataset of deepfake images) and FaceForensics++ [37] (dataset of deepfake videos). Note that videos, in general, are encoded differently than images, and, therefore, the classification performance of the various approaches in the literature degrades dramatically. Despite this, the proposed method works well even on deepfake videos.

The code, models, and datasets are available at the following address: https://iplab.dmi.unict.it/mfs/Deepfakes/MasteringDeepfake2023/.

This article is organized as follows: Section 2 presents state-of-the-art deepfake creation and detection methods. Section 3 and Section 4 describe the dataset and the proposed approach built upon it, respectively. Experimental results and comparison are presented in Section 5. The Section 6 describes the robustness experiments. A dedicated discussion and several hints for future works are given in Section 7. Section 8 concludes the article.

2 RELATED WORKS

As mentioned previously, GAN and Diffusion Models are the main architectures used for synthetic data creation. Section 2.1 will describe some of the most popular deepfake creation architectures based on these two powerful technologies. An overview of the main algorithms for detecting deepfakes on synthetic images created by GAN architectures and the earliest approaches on detecting DM-generated synthetic images will be introduced in Section 2.2.

2.1 Image Synthesis Methods

The most state-of-the-art techniques for creating deepfakes are based on GANs, with several applications on the manipulation of people's faces. StarGAN, proposed by Choi et al. [6], is a framework capable of performing style-based image manipulation (e.g., changing hair color, adding glasses) on multiple domains using a single model. StarGAN-v2 [7], compared with the basic framework (StarGAN), is able to generalize better in the operation of synthesizing the input image, achieving an impressive visual result.

StyleGAN [26] and StyleGAN2 [27] are considered the best engines that can best perform the whole face synthesis operation, resulting in impressive and high-quality photos. The main limitations of StyleGAN consist of the presence of artifacts in the generated synthetic data, which are visible to human eyes. These imperfections have been resolved with StyleGAN2. This framework has been improved with StyleGAN3 [25], a new GAN architecture proposed by Karras et al. The authors addressed the texture sticking problem of GANs in generating videos and animations with 2D transformations. Unlike StyleGAN and StyleGAN2, in StyleGAN3 the number of layers is a free parameter and has no direct relationship with the output resolution.

Deepfakes are not only involved in people face manipulations but they could be engaged to create animals, objects and to solve specific translation tasks such as semantic-to-image operation [29] (e.g., a semantic map is translated in images with realistic details to use in videogames

industry), image-to-image and sketch-to-image translation [48] (e.g., a horse is translated a zebra, a sketch of a black-and-white bag, can be transformed into a colorful, real-life bag).

Advanced in AI technologies opened a new trend in this domain with the Diffusion Model applications. DALL-E 2, proposed by Ramesh et al. [35], presents a new approach for text-conditional image generation. The model leverages the **Contrastive Language-Image Pre-training (CLIP)** framework to encode the text inputs, which are used to condition the generation of the corresponding images. Experimental results show that the proposed model achieves state-of-the-art performance on several text-to-image synthesis benchmarks.

Rombach et al. [36] presented **Latent Diffusion Models (LDMs)**, which perform forward and reverse processes on the latent space learned by an autoencoder. The authors also incorporated cross-attention into the architecture, resulting in further enhancements in conditional image synthesis. The LDM approach was evaluated on super-resolution, image generation, and inpainting tasks. Moreover, the authors demonstrated that LDMs can be trained with limited computational resources, and they significantly reduce inference costs compared to pixel-based diffusion methods.

Proposed by Nichol et al. [33], GLIDE is a novel text-guided diffusion model for photorealistic image generation and editing. The authors use a text encoder to convert textual descriptions into latent vectors, which are then used to guide the image generation process.

2.2 Deepfake Detection Methods

Deepfake detection problems have been addressed by the scientific community with different approaches.

Wang et al. [44] trained a ResNet-50 to discriminate real images from those generated by ProGAN [24] and demonstrated that the trained model is able to generalize for the detection of Deepfakes generated by other architectures than ProGAN. An interesting work known as FakeSpotter was proposed by Wang et al. [43] by describing a new method based on monitoring neuron behaviors of a dedicated CNN to detect faces generated by Deepfake technologies. Several methods work in the Fourier domain with the objective to try to discover discriminative traces on synthetic data left by generative models [18, 31]. The analysis in the Fourier domain was employed by Zhang et al. [46] with AutoGAN in a rather naive strategy that delivered, in any case, good performances. To analytically demonstrate the nature of multimedia content as real or deepfake, Giudice et al. [13] used the **discrete cosine transform (DCT)** to detect so-called **GAN-specific frequencies (GSFs)**, which represent a unique fingerprint of different generative architectures. The β statistics inferred from the distribution of AC coefficients were key to recognizing the data generated by the GAN engine.

There seems to be little work on the detection of synthetic images created by diffusion models. Corvi et al. [8] investigate understanding how difficult it is to distinguish synthetic images generated by diffusion models from real ones and whether current state-of-the-art detectors are suitable for the task. Sha et al. [39] proposed DE-FAKE, a method based on a machine learning classifier for diffusion models detection on four popular text-to-image architectures. The main limitation of this method is that no robustness analysis was performed, and the results are presented only under ideal conditions.

3 DATASET DETAILS

The dataset employed in this study is a dedicated collection of images: real/pristine images collected from CelebA [30], FFHQ,[1] and ImageNet [38] datasets and synthetic data generated by 9 different GAN engines (AttGAN [21], CycleGAN [48], GDWCT [5], IMLE [29], ProGAN [24],

[1] https://github.com/NVlabs/ffhq-dataset

Table 1. Overview of the Images Employed for Training, Validation, and Test Sets (Last Three Columns with the Indication of % of Samples)

Classification Task		Train 50%	Val 20%	Test 30%
14-classes	**Total Images**	28,000	4.200	7.000
	#Img ∀ class	2,000	300	500
13-classes	**Total Images**	26,000	3,900	6,500
	#Img ∀ class	2,000	300	500
L1	**Total Images**	46,480	11,620	24,900
	#Img ∀ class	23,240	5,810	12,450
L2	**Total Images**	23,800	5,950	12,750
	#Img ∀ class	11,900	2,975	6,375
L3 **GANs**	**Total Images**	12.600	3,150	6,750
	#Img ∀ class	1,400	350	750
L3 **DMs**	**Total Images**	11,200	2,800	6,000
	#Img ∀ class	2,800	700	1,500

The first column denotes the classification task (e.g., $14-classes$ is the flat classification task with 14 classes; $L1$ refers to the Level 1 of hierarchy. The $TotalImages$ rows indicate the total number of images employed for training, validation, and testing phases. The $\#Img\forall class$ represents the number of samples considered for each class.

StarGAN [6], StarGAN-v2 [7], StyleGAN [26], StyleGAN2 [27]), and 4 text-to-image DM architectures (DALL-E 2 [35], GLIDE [33], Latent Diffusion [36].[2] For each considered GAN, 2,500 images (a total of 22,500) were generated while for the DMs, 5,000 images were created for each architecture employing more than 800 random sentences, for a total of 20,000 images. Overall, the total number of synthetic data consists of 42,500 images. Finally, for each real dataset (CelebA, FFHQ, and ImageNet) 13,500 images were considered, for a total of 40,500. Table 1 summarizes the numbers of the employed dataset with respect to each level and to the involved splitting of training, validation, and test sets. Figure 3 shows several examples of the obtained dataset. The dataset thus constructed turns out to be novel and challenging not only in terms of semantics (since it contains people's faces, cars, statues, objects of any nature for both real and synthetic data) but presents also variability in terms of resolution, JPEG compression and file format. The deepfake images were generated from the pre-trained models available on the official github repo.[3]

3.1 Preliminary Data Analysis

Building upon previous research in the field, we began by examining the generated dataset, with particular attention given to potential artifacts in the Fourier domain. This approach is crucial for understanding underlying patterns and discrepancies in the data. Figure 4 displays the average

[2]a.k.a. Stable Diffusion: https://github.com/CompVis/stable-diffusion

[3]AttGAN: https://github.com/LynnHo/AttGAN-Tensorflow; CycleGAN: https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix; GDWCT: https://github.com/WonwoongCho/GDWCT; IMLE: https://github.com/zth667/Diverse-Image-Synthesis-from-Semantic-Layout; ProGAN: https://github.com/tkarras/progressive_growing_of_gans; StarGAN: https://github.com/wkentaro/StarGAN; StarGAN-v2: https://github.com/clovaai/stargan-v2; StyleGAN: https://github.com/NVlabs/stylegan; StyleGAN2: https://github.com/NVlabs/stylegan2; DALL-E 2: https://github.com/lucidrains/DALLE2-pytorch; GLIDE: https://github.com/openai/glide-text2im; Latent Diffusion: https://github.com/CompVis/latent-diffusion; Stable Diffusion: https://github.com/CompVis/stable-diffusion

Fig. 3. Examples of images collected from different datasets and images generated by different GANs and DMs.

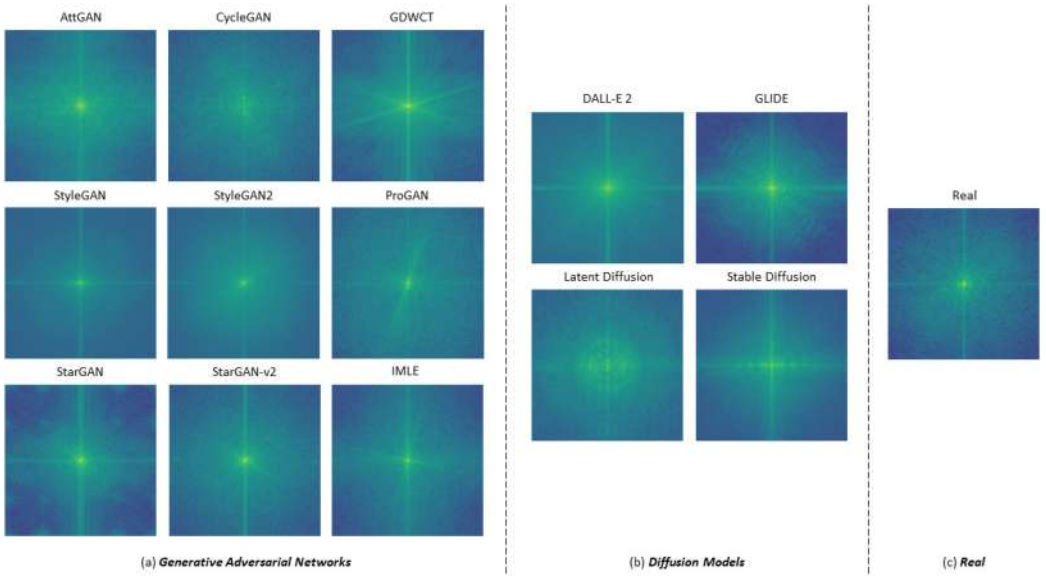


Fig. 4. Average of the Discrete Fourier Transform (DFT) spectrum of the involved data categories: (a) GAN; (b) DM; (c) Real.

Discrete Fourier Transform (DFT) spectrum, calculated across all generated images. The intrinsic signature of GAN-based engines, as documented in earlier studies [18, 31], is evident, suggesting a shared characteristic among these models. Interestingly, DM-based engines appear to be more heavily impacted by noise and visible patterns, which may have implications for their applicability in certain contexts. For comparison and a comprehensive understanding, we also present the average DFT spectrum of images belonging to the "real" class. In this case, the classic isotropic behavior is clearly observable, as expected.

4 MULTI-LEVEL DEEPFAKE DETECTION AND RECOGNITION

The dataset collected and described in the previous section was first investigated as a 14-class[4] "flat" classification task, by employing several deep learning architectures (with 14 classes as output layer).

[4]Total number of classes used in this article.

In addition, a further test was carried out by removing every image belonging to the real class (13 classes as output layer). The trained models showed that in this last case it is possible to achieve greater accuracy score. This gave the idea that a hierarchical approach could lead to even better results giving also a bit of explainability on the analyzed image. The "flat" 14-classes classification task obtained an overall accuracy of only 0, 9571 in the test set with the best architecture. To slightly improve this result, the first class (pristine images) was removed and the model was retrained. With the remaining 13 classes (only synthetic images generated by all considered GAN and DM engines) a bit better overall accuracy of 0, 962 was obtained.

The proposed multi-level deepfake detection and recognition approach consists of 3 levels. Level 1 has the objective to detect real data from those created AI architectures (so, all synthetic data were labeled as belonging to the same class). Given that an image was previously classified as generated by an AI, Level 2 further analyzes images to discriminate between those generated by a GAN from those generated by a DM. Finally, given that an image was previously classified as generated by a GAN or by a DM, the last level solves the task of recognizing the specific architecture between those considered in the dataset. Level 3 is then divided into two sub-modules: "Level 3-GANs" to discriminate the specific GAN across 9-classes; and "Level 3-DMs" to discriminate the specific DM across 4-classes.

All levels were examined by considering different types of CNN architectures, differently trained (with respect to the considered dataset) to solve the specific task. The best E architecture, which performed well in solving all specific problems in the hierarchical approach, was chosen among:

- ResNET-18, ResNET-34, ResNET-50, ResNET-101 [20]: architectures based on the concept of residual blocks;
- ResNEXT-101 [45]: a variant of ResNet that uses multiple convolutional blocks to improve efficiency and performance;
- DenseNET-121 [23]: uses "dense connections" between layers, where each layer receives input from all previous layers, promoting better information sharing between layers;
- EfficientNET-b4, EfficientNET-b7 [42]: optimized architecture that balances model size, depth, and width to achieve high performance with fewer parameters;
- ViT – B16 [11]: a vision transformer architecture, which divides the image into patches and uses self-attention.

Thus, the final architecture is defined by four models: $E_{L_1}, E_{L_2}, E_{L_{3-GAN}}, E_{L_{3-DM}}$. First, an input image I is analyzed and classified by E_{L_1}: $c_{L_1} = E_{L_1}(I), c_{L_1} \in \{real, AI\}$. By following the flow shown in Figure 5(a), if $c_{L_1} = real$, then the analysis of I ends and I is classified as a *real* image. The framework is predisposed to analyze in-depth only in the case where $c_{L_1} = AI$ to discover in detail the nature of the synthetic data under analysis (Figure 5(a)). Thus, when $c_{L_1} = AI$, I is analyzed in Level 2, obtaining the classification $c_{L_2} = E_{L_2}(I), c_{L_2} \in \{GAN, DM\}$ to define the specific AI generative engine (Figure 5(b)). Then, the analysis continues in Level 3-GAN - $c_{L_{3-GAN}} = E_{L_{3-GAN}}(I)$[5] (Figure 5(c)) or in Level 3-DM - $c_{L_{3-DM}} = E_{L_{3-DM}}(I)$[6] (Figure 5(d)) based on the classification results of c_{L_2}. In the latter scenario, it is possible to recognize the specific generative models used to create I.

5 EXPERIMENTAL RESULTS AND COMPARISON

Experiments with all previously listed architectures in all levels and 14- and 13-class classifiers were performed considering the following parameters for training: $batchsize = 30$,

[5] $c_{L_{3-GAN}} \in \{AttGAN, CycleGAN, GDWCT, IMLE, ProGAN, StarGAN, StarGAN - v2, StyleGAN, StyleGAN2\}$.
[6] $c_{3-DM} \in \{DALL - E\ 2, GLIDE, Latent\ Diffusion, Stable\ Diffusion\}$.

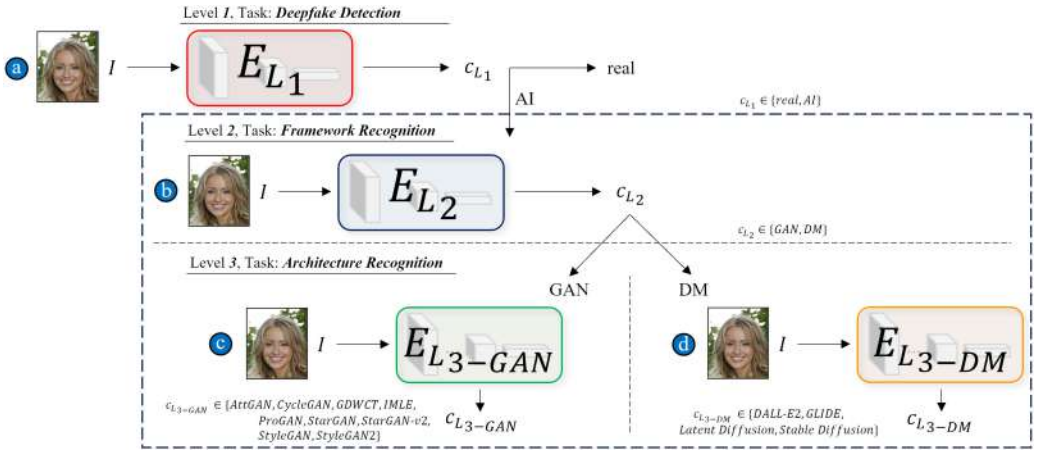


Fig. 5. Detailed sketch of the proposed hierarchical approach. The input image I is first analyzed by model E_{L_1} at level 1 (a). Only in the case where I is classified as generated by AI (c_{L_1} = "*AI*"), the image is analyzed by E_{L_2} in level 2 (b). Thus, if c_{L_2} = "*GAN*", then I is analyzed by $E_{L_{3-GAN}}$: $c_{L_{3-GAN}} = E_{L_{3-GAN}}(I)$ (c) otherwise of $c_{L_{3-DM}} = E_{L_{3-DM}}(I)$ (d) to identify the specific architecture used for creating the deepfakes.

Table 2. Comparison of Average Precision, Recall, f1-Score, and Accuracy Values Obtained among All Involved Architectures (First Column) on All Levels of the Proposed Hierarchical Approach

	Level 1				Level 2				Level 3 - DM				Level 3 - GAN				Overall Framework			
	Prec	Rec	F1	ACC	Prec	Rec	F1	ACC	Prec	Rec	F1	ACC	Prec	Rec	F1	ACC	Prec	Rec	F1	ACC
ResNET-18	0,84	0,98	0,89	0,965	0,95	0,98	0,96	0,9685	0,97	0,97	0,97	0,9716	0,98	0,98	0,98	0,9842	0,97	0,98	0,98	0,9606
ResNET-34	0,88	0,98	0,92	0,9761	0,95	0,97	0,96	0,9629	**0,99**	**0,99**	**0,99**	0,9898	0,98	0,98	0,98	0,9831	0,98	0,98	0,98	0,9621
ResNET-50	0,9	0,98	0,94	0,9818	0,94	0,97	0,95	0,9593	**0,99**	**0,99**	**0,99**	0,986	0,97	0,97	0,97	0,9771	0,97	0,98	0,97	0,9601
ResNET-101	0,94	**0,99**	**0,96**	0,9893	**0,98**	**0,99**	**0,98**	**0,9845**	**0,99**	**0,99**	**0,99**	**0,9937**	0,97	0,97	0,97	0,9701	0,97	**0,99**	0,98	**0,9782**
ResNEXT-101	0,86	0,98	0,91	0,9729	0,93	0,96	0,94	0,9481	0,97	0,97	0,97	0,9707	**0,99**	**0,99**	**0,99**	**0,9925**	0,98	0,98	0,98	0,952
DenseNET-121	0,88	**0,99**	0,92	0,9768	0,96	0,98	0,97	0,9751	**0,99**	**0,99**	**0,99**	0,9861	**0,99**	**0,99**	**0,99**	0,9893	**0,99**	**0,99**	**0,99**	0,972
EfficientNET-b4	0,7	0,93	0,76	0,8996	0,91	0,94	0,92	0,9317	0,79	0,78	0,78	0,7793	0,91	0,92	0,91	0,9141	0,9	0,91	0,9	0,8794
EfficientNET-b7	0,84	0,98	0,89	0,9661	0,91	0,95	0,93	0,935	0,93	0,92	0,92	0,923	0,98	0,96	0,96	0,9808	0,97	0,93	0,95	0,9341
ViT– B16	**0,95**	0,98	**0,96**	**0,9904**	0,92	0,96	0,94	0,945	0,98	0,98	0,98	0,9835	0,98	0,98	0,98	0,9794	0,97	0,98	0,98	0,9533

learningrate = 0.00001, optimizer **Stochastic Gradient Descent (SGD)** with *momentum* = 0.9 and Cross-entropy Loss, epochs = 100. All images were resized to a resolution of 256 × 256, except for ViT, which accepts input images with a size of 384 × 384.

Four different models were obtained for the proposed multi-level deepfake detection and recognition task. Each model was properly trained with a corresponding sub-dataset composed as shown in Table 1. In particular, PyTorch implementations were used of all involved architectures, taking as a starting point the models with the pre-trained weights on ImageNet.[7] For each model, a fully connected layer with an output size equal to the number of classes of the corresponding classification level followed by a SoftMax was added to the last layer of all encoders.

Given the presented solution, four models have to be run simultaneously on a single GPU. This limited the dimension of the useable models.

Experimental results reported in Table 2 have shown that each architecture is able to achieve classification accuracy values greater than 90%. Therefore, defining which turns out to be the best approach for the proposed task is complicated.

We then analyze in detail the various metrics (precision, recall, f1-score, accuracy) to define the best architecture that can solve well the tasks addressed in this article. At level 1, we note that ResNet-101 achieves the best results in recall and f1-score, with precision and accuracy slightly

[7] https://pytorch.org/vision/stable/models.html

lower by 0.01% than the ViT-based model. Excellent results (for all metrics) define the ResNet-101 models the best proposals for levels 2 and 3-DM. Note how the DenseNet-121 architecture manages to achieve performance comparable to ResNet-101 at the 3-DM level and superior to all at the 3-GAN level, with the exception of the ResNEXT-101 model, which also achieves the best accuracy result (0.0032% higher). We note, however, that the ResNET-101 architecture also obtains excellent classification results at this level, with a slight decrease of about 0.02% in all metrics considered. Finally, the overall hierarchical approach (last column of Table 2) shows that both the ResNET-101 and DenseNET-121 models obtain the best results: the first in recall and accuracy and the second in precision, recall, and F1-score. Thus, the ResNET-101 and DenseNET-121 architectures appear to be the best. However, the model based on DenseNET-121 achieves lower performance than ResNET-101, both at level 1 and level 2. The first two levels, as will be described in detail in the following sections, represent those where the misclassification error propagates the most and thus have the greatest weight in the whole hierarchical approach. This turns out to be an extremely crucial element to take into account when choosing the best architecture. Thus, we defined the ResNet-101-based approach as the best solution for all levels.

By using this architecture, it is possible to analyze in detail the data under analysis to reconstruct its history (forensic ballistics on synthetic images); define whether the image is real or manipulated through AI technologies; distinguish well between GAN and DM (in case at the first level it turns out to be generated by AI); define the specific architecture used in the creation procedure. Therefore, although recent CNNs are able to achieve extremely high classification results compared to the basic techniques, in this context, we can therefore state that the configuration of the hierarchical approach based on ResNet-101 turns out to be the best solution ($E = ResNET - 101$) in almost 3 levels both in terms of quality metrics and in terms of forensic ballistics on synthetic data.

Taking advantage of these results, the hierarchical approach described in Section 4 was developed. The best results obtained are the following:

— Level 1: classification accuracy of 0, 9893;
— Level 2: classification accuracy of 0, 9845;
— Level 3: an accuracy of 0, 9701 for the GAN recognition task and an accuracy of 0, 9937 for the DM one.

In general, for each architecture considered in this article, the hierarchical approach compared to the 14-class "flat" classifiers presents a classification accuracy improvement of 2% on average. In the following, all experimental results that we describe will refer to the best chosen architecture: $E = ResNet - 101$.

Figure 6 shows the trend of accuracy and error obtained in the training and testing phases for each epoch and model of the hierarchical approach of levels 1 and 2. In addition, the confusion matrix is shown to further demonstrate the potential of the proposed approach. Similarly, Figure 7 shows the accuracy and error performance of level 3, both for the recognition task of the specific GAN architecture that generated the synthetic data with respect to 9 classes and for the recognition task of the specific DM architecture that generated the deepfake image with respect to 4 classes.

5.1 Flat and Hierarchical Approach in Comparison

As shown in Table 2, the classification results obtained with the hierarchical approach turn out to be better than the 0, 9571 of the 14-class flat approach, as each level obtains an accuracy over 97%. However, it is necessary to analyze these results in detail, since they refer to accuracy values of individual levels (as if they were independent blocks of each other) and do not take into account the cumulative error of the entire framework. The results obtained show only that, by considering a different number and organization of classes at each level, the classification accuracy results

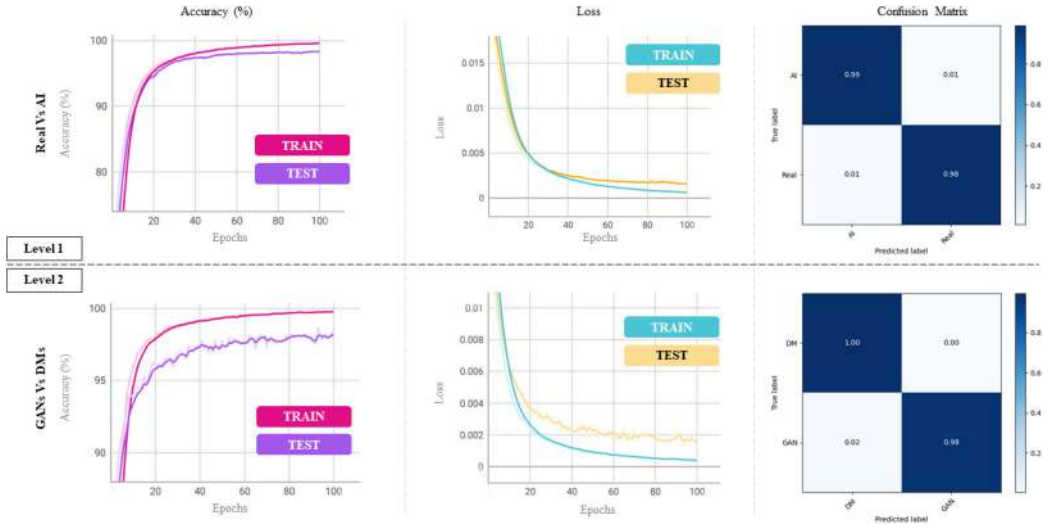


Fig. 6. Trend of accuracy (%) and loss values obtained in the training and testing phases for each epoch of level 1 (Real vs. AI) and level 2 (GANs vs. DMs) of the proposed hierarchical approach. Confusion matrices for the involved classification tasks were reported.

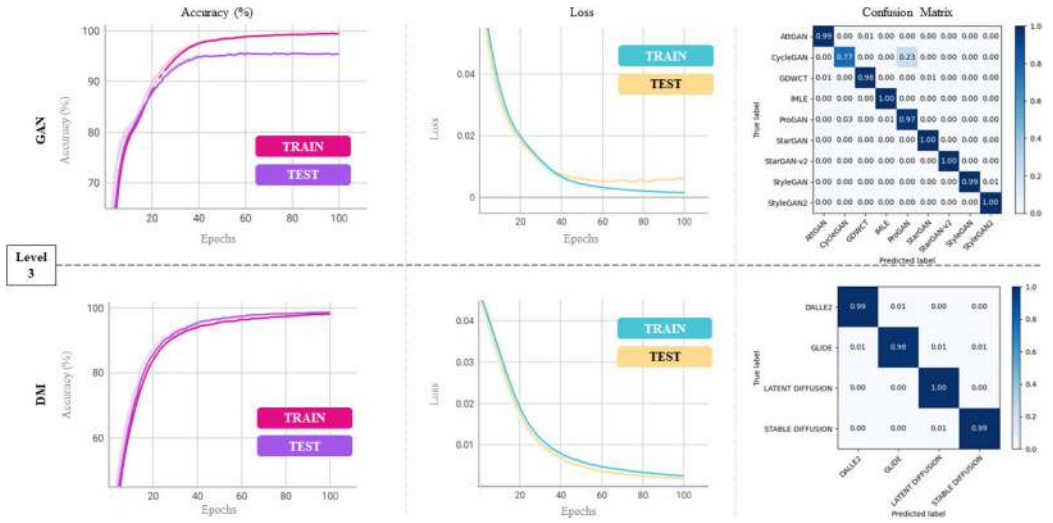


Fig. 7. Trend of accuracy (%) and loss values obtained in the training and testing phases for each epoch of level 3 of the hierarchical approach. Specifically, the first row refers to the recognition task of the specific GAN architecture (with respect to 9 classes) that generated the synthetic data. The second row refers to the recognition task of the specific DM architecture (with respect to 4 classes) that generated the synthetic data. In addition, confusion matrices for the involved tasks were reported.

improve. It is therefore necessary to take into account the cumulative error between levels to define and demonstrate in the whole, whether the hierarchical approach performs better than the flat method. Let I be an image of real class given as input to the hierarchical approach. At level 1, if I is classified as real, then the execution flow would end (so, there would be no cumulative error to propagate and consider in the next levels). In the case where I is misclassified as belonging to the

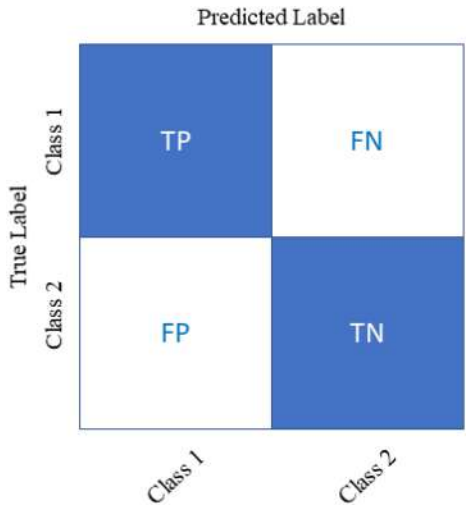


Fig. 8. Example of confusion matrix between Class 1 and Class 2. TP, TN, FP, FN represent the True Positive, True Negative, False Positive, and False Negative, respectively.

AI class, then the image under analysis will also be analyzed by all further levels, resulting in more misclassifications. In this context, the framework will misclassify the image 3 times (one error for each level), and in the final accuracy calculation, all these errors must be counted. Consider the scenario where the input image I is deepfake and created by a GAN architecture. At the first level, if I is classified as belonging to the real class, then the error will be counted but not propagated to subsequent levels, because the execution flow would terminate. These errors will need to be counted in the overall accuracy calculation, however. In case I, at Level 1, is classified correctly (belonging to class *AI*), no error will be propagated to subsequent levels, since the classification is correct. The image then, following the proposed hierarchical pipeline, will be analyzed at Level 2: In the case of correct classification (belonging to the GAN class), no error will be propagated to Level 3 − *GAN*. Conversely, if at Level 2 the image is classified as belonging to the class *DM*, then the classification will be incorrect and inevitably also at Level 3 − *DM*. In this context, the framework will misclassify the image two times (one error at each Level 2 and one error at Level 3) and in the final accuracy calculation, all these errors must be counted. At level 3, considering the case that all classifications in the previous levels are correct, the specific architecture that generated the synthetic data will be defined (Deepfake Architecture Recognition). In the positive case, the execution flow of the hierarchical approach would end with all Deepfake Detection and Recognition tasks resolved correctly. In case of wrong architecture recognition, the error at level 3 should be counted and accounted for in the overall accuracy of the hierarchical approach. It is also clear that all correct classifications must also be counted in the total. Let us consider the generic schema of a confusion matrix, as shown in Figure 8, where

— True Positive (TP) True Positive (TP): event in which the predicted class is equal to the actual class;
— True Negative (TN): event in which the predicted class is equal to the actual class (different from the TP class);
— False Positive (FP): event in which the predicted class is equal to the TP class but is different from the actual class;
— False Negative (FN): event in which the predicted class is equal to the class of TN but is different from the actual class.

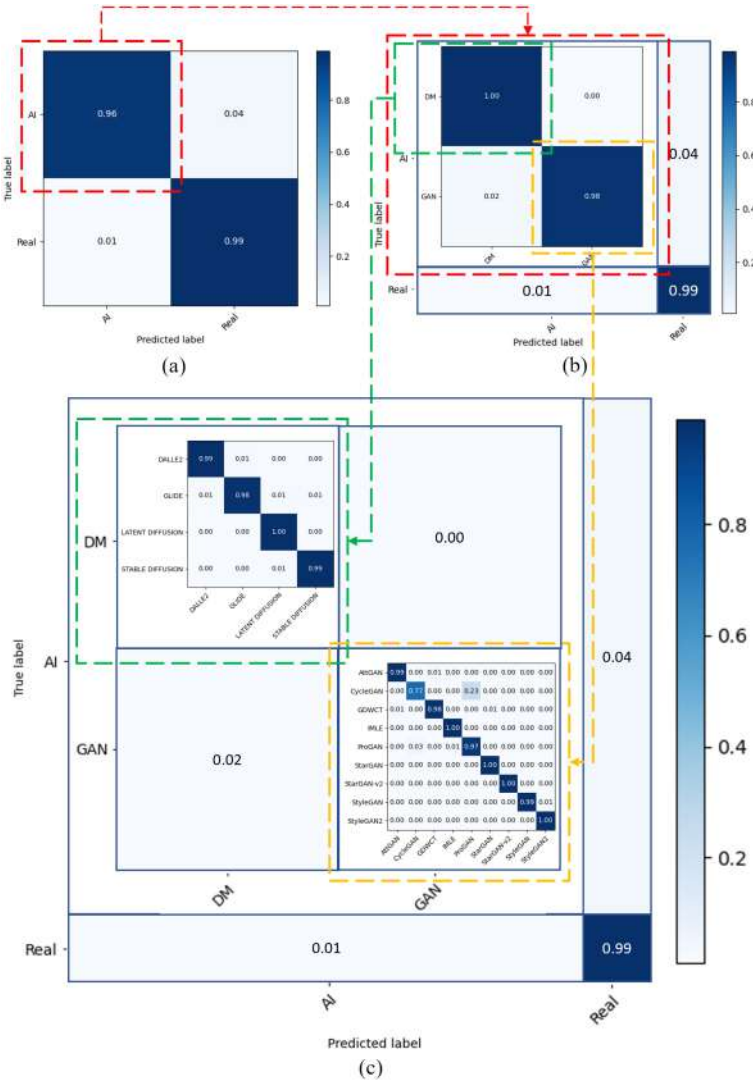


Fig. 9. Overall confusion matrix of the hierarchical approach: (a) Confusion matrix of Level 1. The True Positive (related to AI class) in (a) can be replaced as shown in (b), which represents the classification accuracy at Level 2. Similarly, the True Positives (DM class) and True Negatives (GAN class) in (b) can be replaced with (c), which describes the classification accuracy at Level 3. Through (c), the total classification accuracy of the hierarchical approach can be calculated (applying Equation (1)) by accumulating the classification error at each level.

The classification accuracy value is given by Equation (1):

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}. \tag{1}$$

To better understand the propagation and accumulation of error and correct classifications among the various levels of the hierarchical approach, let us consider the subsequence of confusion matrices in Figure 9.

Table 3. Comparison with State-of-the-art Approaches

	Level 1	Level 3	
	Real vs. AI	**GANs**	**DMs**
AutoGAN [46]	0,685	0,803	—
Fakespotter [43]	0,7422	0,9532	—
EM [16]	0,8657	0,9502	—
DCT [13]	0,8720	0,9589	—
Wang et al. [44]	0,7854	0.9732	—
DE-FAKE [39]	0,9052	–	0,9345
Our	**0,9893**	**0,9701**	**0,9937**

Classification accuracy value is reported. — are reported where the method can not handle the corresponding task.

At level 1 of the hierarchical approach, we will have the scheme in Figure 9(a). The TP block (referring to the AI class) at level 1 can be "mapped" to level 2 as shown in Figure 9(b) with an additional confusion matrix. Similarly, the TP and TN blocks of level 2 can be mapped to the level with corresponding confusion matrices as shown in Figure 9(c). To consider the error propagation and correct classifications of the various levels for calculating the classification accuracy of the whole hierarchical approach, all the results in Figure 9(c) should be counted.

The cumulative error counting was done by considering the following scheme:

— If the image from level 1 is misclassified as belonging to the AI class, then this error will be counted 3 times (which will represent one error for each level). If the image is misclassified as belonging to the real class, as previously described, then the error will be counted only once.
— If the image is misclassified from level 2, then this error will be counted twice (which will represent an error for level 2 an error for level 3).
— If the image is misclassified by level 3, then the error will be counted only once.

Pseudocode 1 summarizes how the accuracy value is calculated in this context.[8] Also note that by applying Equation (1) to calculate classification accuracy, considering all correct and misclassification values in Figure 9(c), we obtain the same value as Pseudocode 1.

The total classification accuracy of the hierarchical approach is 0, 9782. Compared with the flat method (accuracy = 0, 9571), the performance is not only higher, but with the hierarchical approach, we have the ability to better understand and discriminate the nature of the data under analysis.

5.2 Comparison with State-of-the-art Methods

The best models ($E_{L_1}, E_{L_2}, E_{L_{3-GAN}}, E_{L_{3-DM}}$ with $E = ResNET - 101$) of the hierarchical approach were compared with various state-of-the-art works such as References [13, 16, 39, 43, 44, 46], demonstrating to achieve best results in each task. For each method, a pretrained model was taken into account and a fine-tuning operation was performed considering our training datasets.

We highlight that, in general, not all state-of-the-art methods are set up to solve all the classification tasks described in this article. A generic state-of-the-art method might be re-trained to solve a different task, but the latter might be further penalized because it was prepared, designed, and calibrated to solve a specific task. For this reason, we made a point-to-point comparison with various state-of-the-art methods devoted to solve the same specific task (of the level under consideration).

[8]In the same way precision, recall, and f1-score metrics are calculated.

ALGORITHM 1: Pseudocode for calculating the total accuracy of the hierarchical framework. *TestSet* represents the dataset of the Test Set. I represents a generic image from the Test set. GT represents the Ground Truth (defined on the 14 classes: $\{AttGAN, CycleGAN, \ldots, Real\}$) of I. The $map(GT)$ function returns the GT label mapped to the level it belongs to. At each level, the total error is counted in case of incorrect classification (*error* variable), otherwise, the correct classification is counted (*correct* variable). The *accuracy* variable will contain the final accuracy value of the hierarchical approach. Note that at level 3, image I will be analyzed by either the *GAN* module or the *DM* module. Counting error and classification correctness is done in the same way for both modules.

```
Data: TestSet
Result: accuracy
error ← 0;
correct ← 0;
for I, GT in TestSet do
    target ← map(GT,'L1');                                  /* *** LEVEL 1 *** */
    pred ← Level1(I);
    if pred != target && pred == "real" then
        error ← error + 1;
    else
        if pred != target then
            error ← error + 3;
        else
            correct ← correct + 1;
            target ← map(GT,'L2');                          /* *** LEVEL 2 *** */
            pred ← Level2(I);
            if pred != target then
                error ← error + 2;
            else
                correct ← correct + 1;
                target ← map(GT,'L3');                      /* *** LEVEL 3 *** */
                pred ← Level3(I);
                if pred != target then
                    error ← error + 1;
                else
                    correct ← correct + 1;
                end
            end
        end
    end
end
accuracy ← correct/(error + correct);
```

In Table 3, the Level 3 DMs column does not show some values given that corresponding approaches does not cover the corresponding task. Therefore, most of the methods do not seem to be able to effectively distinguish between different DMs. This is because DMs leave different traces on synthetic images than those generated by GAN engines. This claim is further empirically demonstrated by the results reported in Table 3 on the "Real vs. AI" column, where, due to the presence of DMs-generated images, the classification accuracy values of the various methods are sensibly lower than the 0, 9893 obtained by the proposed approach. For this reason, a column

regarding Level 2 task was not added, as the classification results of the various methods turn out to be significantly lower (almost random classifiers) than the classification accuracy of 0, 9845% obtained by the proposed approach. Finally, all state-of-the-art methods, including the proposed one, are able to generalize well in terms of distinguishing between GAN architectures that created the synthetic data (GANs column).

As previously described, not all state-of-the-art methods are predisposed to solve all tasks in the hierarchical approach. A level-by-level comparison with the various architectures would mean altering (even minimally) the state-of-the-art method with which we compare by adding neural layers to solve the task in question. However, this could alter the very nature of the state-of-the-art method (since it was designed and set up ad hoc to solve that specific task) and would be further penalized. For this reason, we made other comparisons by considering other state-of-the-art methods implemented to solve that specific task in such a way that the comparison with some of the levels of the hierarchical approach turns out to be fair. We compared Level 3 - DM with the Reference [15] method, which is similar to Reference [44], but it uses modified Resnet50 and is trained on StyleGAN images. Also for this method, the pretrained model was considered and a fine-tuned operation was performed with our dataset. The Level 3 - DM succeeds in obtaining a higher accuracy result (0, 9937) than the 0.929 of Reference [15]. A more recent method in this context is DIM [40], which is able to extract fingerprints from generated images to identify the architecture that generated the synthetic data. Performing training from scratch with the training data of Level 3 - DM, an accuracy value of 0.8965 is obtained. The latter method was also compared with Level 3 - GAN, obtaining a classification accuracy value of 0.9518. Even in this context, the Level 3 - GAN block manages to perform better with a value of 0, 9701.

6 ROBUSTNESS AND GENERALIZATION EXPERIMENTS

To demonstrate the generalizability of the proposed hierarchical approach, several robustness tests were performed. In particular, considering images from test sets (never used in the training procedure), the following attacks were applied:

(1) *Gaussian blur* with kernels of different sizes (3×3, 9×9);
(2) *Rotation* of 45, 135, 225, 315 degrees: rotations implement interpolation operations to add new information (if necessary);
(3) *Resize* of +50%, −50%: due to the interpolation operations performed, information is added or removed, respectively;
(4) *JPEG Compression* with quality factor of {50, 60, 70, 80, 90}: Lossy compression removes high-frequency information that could drastically alter the performance of any classifier. In this context, the lower the Quality Factor value, the more information is removed.

Figure 10 shows images (one for each class) obtained after applying the filters listed above. Each of these manipulations could destroy those intrinsic features/traces in the synthetic images left by the various generative models during the creation process. It turns out to be important to demonstrate how robust the proposed approach is to this type of attack to counter the illicit use of these powerful generative models. Therefore, the various models characterizing the hierarchical approach (trained only and exclusively with RAW images, i.e., images without attacks) were tested by considering all the images in Figure 10. The classification results are reported in Figure 11. The average classification accuracy value of each involved attack is reported in Figure 12.

As can be seen, the results obtained from the robustness test show that the hierarchical model (not trained with these manipulated images) is robust to various kinds of attacks. Gaussian blur is the result of blurring an image using a Gaussian function. Using a 3×3 kernel size, an average classification accuracy for all 3 levels of 0.839 is obtained. The larger a kernel size is considered,

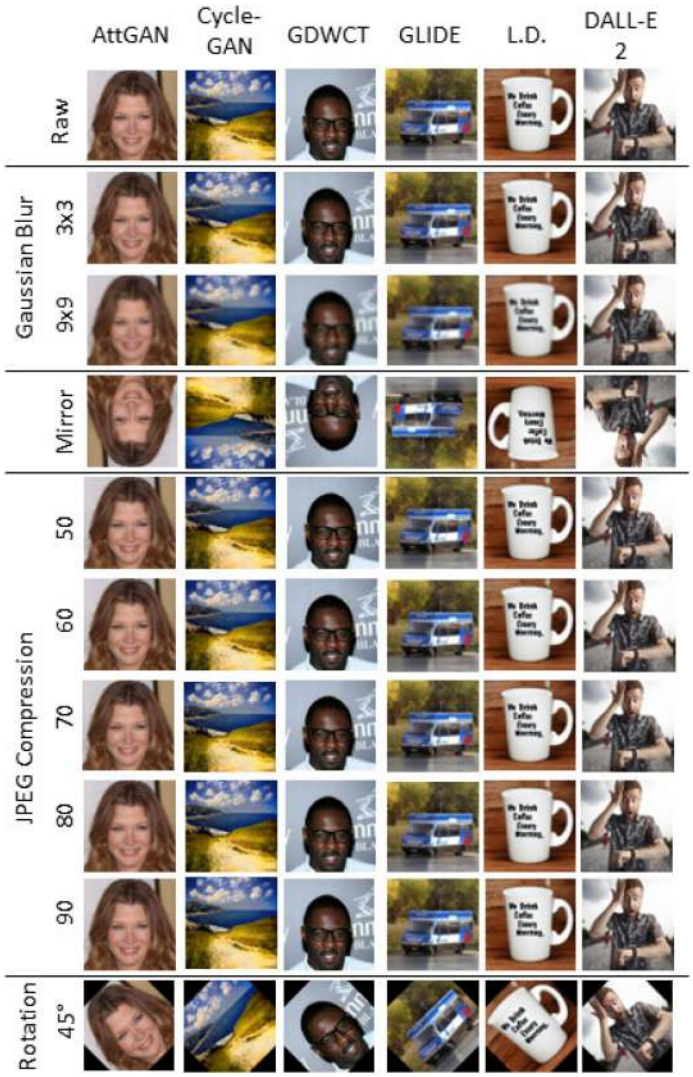
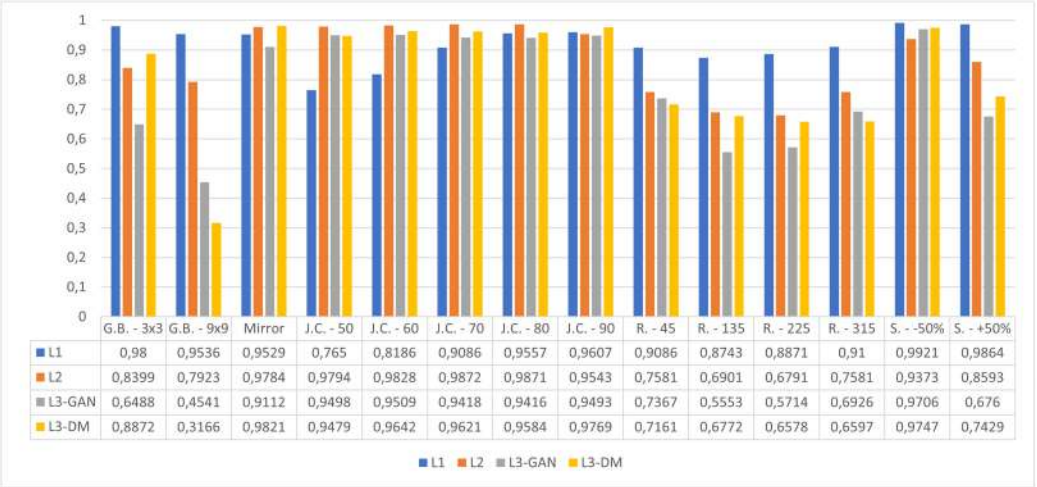


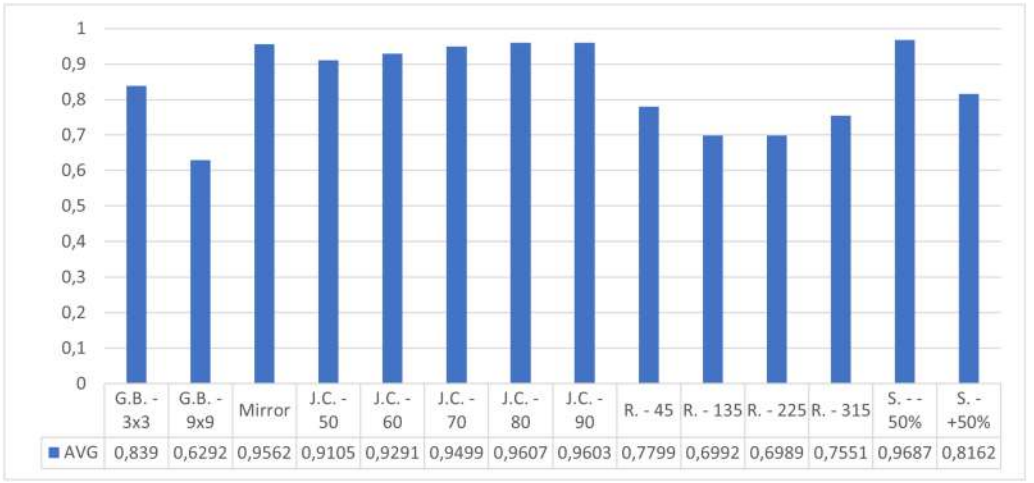
Fig. 10. Example of images to which some of the robustness attacks have been applied. Note that not all the types of images (e.g., StarGAN, StyleGAN, real) and manipulations are reported (e.g., for rotation, only the example with rotation with degree 45 was reported; no example was reported for the resize operation).

the blurrier and more degraded the final signal results. The classification accuracy collapses dramatically and mainly on images created by DM. This could be mainly due to the very nature of how these data are generated (see the creation process in Figure 1). Probably the added Gaussian noise has destroyed that pattern left by DM engines; a pattern/signature learned during the training process by the gradual addition of Gaussian noise and inserted into the multimedia data by the generation process. In application contexts, particularly in the case of cyber attacks, it is rare to find blurred images with Gaussian noise with a high kernel size, as the signal itself would be extremely degraded. The model fails to generalize well with the rotation operation, as, on all three levels, an average classification accuracy value of 0.7332 is obtained. The rotation operation not only applies interpolation to fill in the missing information, but also draws the (x,y) coordinate



	G.B. - 3x3	G.B. - 9x9	Mirror	J.C. - 50	J.C. - 60	J.C. - 70	J.C. - 80	J.C. - 90	R. - 45	R. - 135	R. - 225	R. - 315	S. - -50%	S. - +50%
L1	0,98	0,9536	0,9529	0,765	0,8186	0,9086	0,9557	0,9607	0,9086	0,8743	0,8871	0,91	0,9921	0,9864
L2	0,8399	0,7923	0,9784	0,9794	0,9828	0,9872	0,9871	0,9543	0,7581	0,6901	0,6791	0,7581	0,9373	0,8593
L3-GAN	0,6488	0,4541	0,9112	0,9498	0,9509	0,9418	0,9416	0,9493	0,7367	0,5553	0,5714	0,6926	0,9706	0,676
L3-DM	0,8872	0,3166	0,9821	0,9479	0,9642	0,9621	0,9584	0,9769	0,7161	0,6772	0,6578	0,6597	0,9747	0,7429

Fig. 11. Classification accuracy of robustness test (G.B. = Gaussian Blur, J.C. = JPEG Compression, R. = Rotation, S = Scaling).



	G.B. - 3x3	G.B. - 9x9	Mirror	J.C. - 50	J.C. - 60	J.C. - 70	J.C. - 80	J.C. - 90	R. - 45	R. - 135	R. - 225	R. - 315	S. - - 50%	S. - +50%
AVG	0,839	0,6292	0,9562	0,9105	0,9291	0,9499	0,9607	0,9603	0,7799	0,6992	0,6989	0,7551	0,9687	0,8162

Fig. 12. Average Accuracy Value of the robustness tests of the three levels of the hierarchical approach (G.B. = Gaussian Blur, J.C. = JPEG Compression, R. = Rotation, S = Scaling).

pixels into new (x',y') coordinates. Inevitably, in this context, the pattern left by that generative model is all but destroyed. However, one key element should be noted. The rotations applied in this section will add black bands in the resulting image. If we were to consider a real context, then this type of attack with high probability will never be used, since the media content in question will present very different characteristics to real images (the human eye would be able to define the content as manipulated). Despite this, the model still manages to achieve a performance above 70% average classification accuracy. Surprising are the results with JPEG compression attacks, with accuracy values almost all over 91%, with an average of 0.9421 considering all levels. The method also turns out to be robust to resize attacks, with an average accuracy across the three levels of 0.8924. These results show that the proposed hierarchical approach can be applied in real-world contexts:

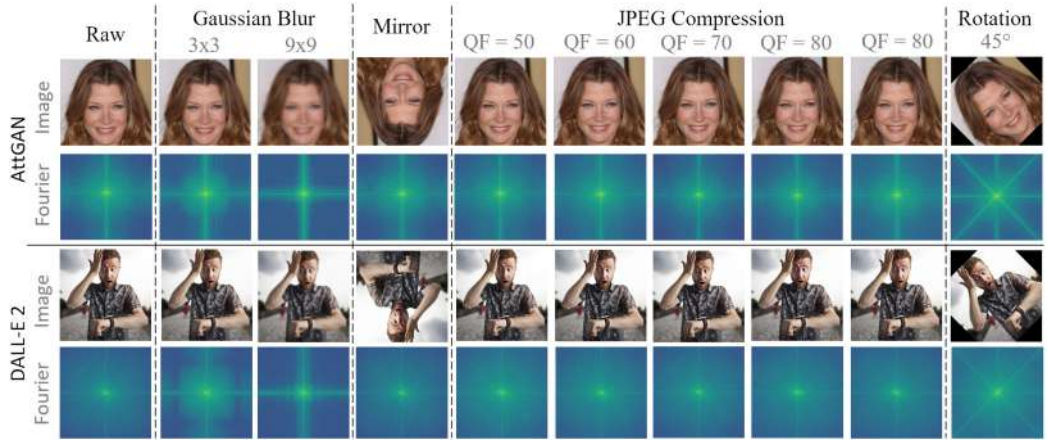


Fig. 13. Fourier spectra computed on images created from GAN (AttGAN) and DM (DALL-E 2) generative models after applying the robustness filters. It can be seen that in many cases such fits greatly alter the frequencies of the input signal.

Facebook, Instagram, WhatsApp apply JPEG compressions (with quality factor 50) and resizing operations. In addition, Mirroring is one of the filters available in applications such as Instagram. In this case, the three layers on average achieve a classification accuracy value of 0.9561.

The results obtained from the robustness tests also show that no data augmentation operation is needed in the learning phase in the various levels of the hierarchical approach.

We also want to highlight the fact that the images used during training (done only once considering only Raw images) are not only not subject to the attacks described in this section, but are scaled to a resolution of 256×256. The results obtained show that this operation does not affect the classification performance of the various levels.

Generalization. To demonstrate the generalization capability of the proposed models, we tested the $E_{L_1}, E_{L_2}, E_{L_{3-GAN}}, E_{L_{3-DM}}$ with $E = ResNET - 101$ by considering the following dataset (not considered in the dataset described in Section 3):

— COCOfake [2]: Among the most recent multimodal dataset on deepfakes, it is composed of images optimally manipulated through the diffusion model Stable Diffusion, starting from COCO dataset.
— FaceForensics++ (FF++) [37]: A dataset of deepfake videos.

With regard to FF++, only videos manipulated via deepfake technologies were chosen, i.e., those belonging to the DeepFakes and NeuralTextures category. In addition, to classify a dataset as deepfake, for each video, faces (identified through the DLIB library) were extracted from the first 10 video frames and classified individually. Good generalization results were obtained with both datasets (Table 4):

The COCOfake dataset can be analyzed by all 3 levels of the hierarchical approach, as the architecture at level 2 provides the Diffusion model category and at level 3 provides the stable diffusion category. It is not possible to do the same for FF++, since different technologies are used for creating deepfake videos. Therefore, FF++ was tested only for level 1. The approach then used to define a video (from FF+) as a deepfake was to classify the individual frame (in our case, we analyze only the first 10). This approach could be useful in the case where an attacker manipulates only a subset of frames; it is then sufficient to find the first frame attacked to assert manipulation of the video itself. In the case of FF++, it is known *a priori* that a deepfake video has all the frames manipulated;

Table 4. Generalization Test Performed on the COCOFake [2] and FaceForensics++ (FF++) [37] Datasets

ACC	L1	L2	L3-DM	Overall
COCOFake [2]	0,9197	0,9302	0,7587	0,8538
FF++ [37]	0,94	–	–	–

The classification accuracy value (ACC) is reported for each involved experiment.

therefore, we chose only to analyze the first 10 frames and work only on the face regions: The manipulation is mainly applied in those areas with respect to the background. We want to highlight how well our method can also work in the context of videos. Videos are encoded differently than images, so the method could also achieve much lower accuracy values. In general, however, to be able to generalize better even with videos, as future works, we will set up the framework to work optimally by defining the best strategy: For example, one could work by considering only key frames and through voting techniques define whether the video under analysis is deepfake.

7 DISCUSSION AND FUTURE WORKS

The ResNET-101-based architecture succeeds in generalizing well, even in contexts where the image is attacked (as demonstrated in Section 6), in classification and architecture recognition tasks by considering data of different semantics and dataset types never seen before, such as COCOFake (deepfake images) and FaceForensics++ (deepfake videos). We want to highlight the fact that we manage to generalize even with videos that, in general, have different encoding and compression than images; note that videos were never used during the training procedure of ResNET-101. Considering datasets of different nature (in terms of semantics) and type, the performance of the methods with which we compare degrade [13, 43, 46]. For example, the GAN-DCT [13] method manages to generalize well with images, but on deepfake videos the performance decreases. This is mainly because the β-statistics extracted from images turn out to be different in videos, precisely because of the specific nature of the type of data under examination. The proposed hierarchical approach, despite the excellent results obtained even with digital videos, could be extended and improved by defining new strategies: For example, in the case of videos, only key frames could be considered, and through voting techniques, the nature of the video itself could be asserted.

Most of the approaches available in the state-of-the-art have different generalization and robustness capabilities, but some have specific limitations. For example, Reference [46] is a method that works mainly in the frequency domain. A simple attack based on JPEG compression [4] could destroy those intrinsic traces, i.e., those discriminative frequencies, causing a decrease in classification performance. The proposed models of the three levels of the hierarchical approach turn out to be robust to these attacks. In addition, almost all the methods we deal with are also limited by the fact that they use data containing the same semantics [43], i.e., people's faces. However, References [13, 46] turn out by their nature to be necessarily constrained to the specific semantics, as References [12, 47] demonstrated that those specific features are used to discriminate the scene itself. Therefore, in the case of multimodal data analysis, different state-of-the-art methods would focus more on semantics than on the traces left by generative models during the creation process. In the latter scenario, such methods would not be able to generalize in solving specific deepfake detection tasks. Instead, the method cited in Reference [44] seems to be the most promising one for the purpose of achieving generalization. Unfortunately, however, as the number of generative architectures (such as those used in this work) increases, the classification performance degrades. This mainly happens with images generated by diffusion models, as they tend to have statistics very similar to real data. In general, however, it is possible to say that state-of-the-art methods

work well under specific configurations, and to achieve generalization they should be refactored and improved to solve the specific limitations [1, 10, 28].

Moreover, state-of-the-art architectures mainly focus on solving the classical Real vs. Deepfake binary classification task, resulting to be not very useful in forensic applications. The proposed method overcomes these limitations and turns out to be able to analyze the multimedia data even more in detail: Level by level, we define not only the technology used between GAN and DM but we manage, with high results, to define the specific architecture used to create the synthetic data. Levels 2 and 3 can be reprojected to generalize more by considering frameworks and architectures even not known *a priori*, exploiting the potential of approaches based on the concept of clustering and metric learning.

In addition, the models $E_{L_1}, E_{L_2}, E_{L_{3-GAN}}, E_{L_{3-DM}}$ are based, in this work, on the same $E = ResNET - 101$ architecture. Future works will focus on exploring a combination of the different engines that performed best in solving their respective tasks (Table 2)[9] to improve the performance of the overall framework.

Finally, the hierarchical approach can be further extended by adding an additional level (level 4) that can identify the specific instance of the architecture used [19], i.e., defining the specific model (defined as the set of its parameters) used for the creation of the synthetic data (Deepfake Model Recognition).[10]

8 CONCLUSIONS

In this article, a deepfake detection and recognition solution has been presented. The proposed solution is capable of recognizing whether an image was generated using 9 different GAN engines and 4 diffusion models (DMs) by means of a hierarchical approach. Furthermore, the proposed solution has demonstrated superior performance over the state-of-the-art in all addressed tasks. The models obtained in the different levels turn out to be robust to various attacks such as JPEG compression (with different quality factor values) and resize, demonstrating that the framework can be used and applied in real-world contexts to counter the illicit use of these powerful and modern generative models. One of the primary challenges faced in the field is the increasing prevalence of false content generated through machine learning techniques, such as Generative Adversarial Networks (GANs) and DMs. These artificial intelligence algorithms have reached remarkable levels of sophistication, making it increasingly difficult to distinguish between real and fabricated content. The need to develop effective methods for the detection and recognition of these deepfakes has thus become crucial to ensure information integrity and protect individuals' privacy. In the context of this research, we developed a deepfake detection and recognition system based on the analysis of distinctive characteristics of images generated by different GAN and DMs engines by using standard deep learning architectures. Following this strategy, we were able to identify the intrinsic peculiarities of images generated by these algorithms, allowing our system to classify them with high accuracy. The proposed solution has demonstrated high recognition capabilities, surpassing the state-of-the-art in all addressed tasks. Finally, preliminary results show that the combination of the best architectures (E_{L1} = ViT-B16, E_{L2} = ResNET-101, E_{L3-GAN} = DenseNET-121, E_{L3-DM} = ResNET-101) manage to achieve an average overall classification accuracy of 98.21%. In future work, to further improve the performance of the hierarchical approach, increasingly

[9]For instance, it is possible to consider $E_{L_1} = ViT$, $E_{L_2} = ResNET - 101$, $E_{L_{3-GAN}} = ResNEXT - 101$, $E_{L_{3-DM}} = ResNET - 101$.

[10]For example, given the StyleGAN architecture, it is possible to retrain a base model (even by changing a simple value of a weight) to obtain a new StyleGAN instance. Similarly, it is possible to define N instances of StyleGAN. The task of Deepfake Model Recognition is to define the specific instance among the N.

high-performance deep architectures will be considered. The framework including the models and code is available at https://iplab.dmi.unict.it/mfs/Deepfakes/MasteringDeepfake2023/.

REFERENCES

[1] Lydia Abady, Jun Wang, Benedetta Tondi, and Mauro Barni. 2023. A siamese-based verification system for open-set architecture attribution of synthetic images. *arXiv preprint arXiv:2307.09822* (2023).

[2] Roberto Amoroso, Davide Morelli, Marcella Cornia, Lorenzo Baraldi, Alberto Del Bimbo, and Rita Cucchiara. 2023. Parents and children: Distinguishing multimodal deepfakes from natural images. *arXiv preprint arXiv:2304.00500* (2023).

[3] Sebastiano Battiato, Oliver Giudice, and Antonino Paratore. 2016. Multimedia forensics: Discovering the history of multimedia contents. In *Proceedings of the 17th International Conference on Computer Systems and Technologies*. 5–16.

[4] Sebastiano Battiato, Massimo Mancuso, Angelo Bosco, and Mirko Guarnera. 2001. Psychovisual and statistical optimization of quantization tables for DCT compression engines. In *Proceedings of the 11th International Conference on Image Analysis and Processing*. IEEE, 602–606.

[5] Wonwoong Cho, Sungha Choi, David Keetae Park, Inkyu Shin, and Jaegul Choo. 2019. Image-to-image translation via group-wise deep whitening-and-coloring transformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10639–10647.

[6] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8789–8797.

[7] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020. StarGAN v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8188–8197.

[8] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. 2023. On the detection of synthetic images generated by diffusion models. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'23)*. IEEE, 1–5.

[9] Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat GANs on image synthesis. *Adv. Neural Inf. Process. Syst.* 34 (2021), 8780–8794.

[10] Shichao Dong, Jin Wang, Renhe Ji, Jiajun Liang, Haoqiang Fan, and Zheng Ge. 2023. Implicit identity leakage: The stumbling block to improving deepfake detection generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3994–4004.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, and Sylvain Gelly. 2020. An image is worth 16 × 16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*.

[12] Giovanni Maria Farinella, Daniele Ravì, Valeria Tomaselli, Mirko Guarnera, and Sebastiano Battiato. 2015. Representing scenes for real-time context classification on mobile devices. *Pattern Recognition* 48, 4 (2015), 1086–1100.

[13] Oliver Giudice, Luca Guarnera, and Sebastiano Battiato. 2021. Fighting deepfakes by detecting GAN DCT anomalies. *Journal of Imaging* 7, 8 (2021), 128. https://doi.org/10.3390/jimaging7080128

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Adv. Neural Inf. Process. Syst.*. 2672–2680.

[15] Diego Gragnaniello, Davide Cozzolino, Francesco Marra, Giovanni Poggi, and Luisa Verdoliva. 2021. 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 (ICME)*. IEEE, 1–6.

[16] Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. 2020. Fighting deepfake by exposing the convolutional traces on images. *IEEE Access* 8 (2020), 165085–165098. DOI: https://doi.org/10.1109/ACCESS.2020.3023037

[17] Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. 2022. Deepfake style transfer mixture: A first forensic ballistics study on synthetic images. In *International Conference on Image Analysis and Processing (Lecture Notes in Computer Science,* Vol. 13232*)*. Springer, Cham, 151–163.

[18] Luca Guarnera, Oliver Giudice, Cristina Nastasi, and Sebastiano Battiato. 2020. Preliminary forensics analysis of deepfake images. In *Proceedings of the AEIT International Annual Conference (AEIT'20)*. IEEE, 1–6. https://doi.org/10.23919/AEIT50178.2020.9241108

[19] Luca Guarnera, Oliver Giudice, Matthias Nießner, and Sebastiano Battiato. 2022. On the exploitation of deepfake model recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 61–70.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[21] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. AttGAN: Facial attribute editing by only changing what you want. *IEEE Trans. Image Process.* 28, 11 (2019), 5464–5478.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* 33 (2020), 6840–6851.
[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4700–4708.
[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive growing of GANs for improved quality, stability, and variation. In *Proceedings of the International Conference on Learning Representations (ICLR'18)*.
[25] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Adv. Neural Inf. Process. Syst.* 34 (2021), 852–863.
[26] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4401–4410.
[27] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8110–8119.
[28] Chuqiao Li, Zhiwu Huang, Danda Pani Paudel, Yabin Wang, Mohamad Shahbazi, Xiaopeng Hong, and Luc Van Gool. 2023. A continual deepfake detection benchmark: Dataset, methods, and essentials. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1339–1349.
[29] Ke Li, Tianhao Zhang, and Jitendra Malik. 2019. Diverse image synthesis from semantic layouts via conditional IMLE. In *Proceedings of the IEEE International Conference on Computer Vision*. 4220–4229.
[30] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*. 3730–3738.
[31] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. 2019. Do GANs leave artificial fingerprints? In *Proceedings of the IEEE Conference on Multimedia Information Processing and Retrieval (MIPR'19)*. 506–511.
[32] Momina Masood, Mariam Nawaz, Khalid Mahmood Malik, Ali Javed, Aun Irtaza, and Hafiz Malik. 2023. Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. *Applied intelligence* 53, 4 (2023), 3974–4026.
[33] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. 2022. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the International Conference on Machine Learning*. PMLR, 16784–16804.
[34] Ehsan Nowroozi, Ali Dehghantanha, Reza M. Parizi, and Kim-Kwang Raymond Choo. 2021. A survey of machine learning techniques in adversarial image forensics. *Comput. Secur.* 100 (2021), 102092.
[35] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).
[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10684–10695.
[37] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. 2019. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1–11.
[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. 2015. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115 (2015), 211–252.
[39] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. 2022. DE-FAKE: Detection and attribution of fake images generated by text-to-image diffusion models. *arXiv preprint arXiv:2210.06998* (2022).
[40] Sergey Sinitsa and Ohad Fried. 2023. Deep image fingerprint: Accurate and low budget synthetic image detector. *arXiv preprint arXiv:2303.10762* (2023).
[41] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2256–2265.
[42] Mingxing Tan and Quoc Le. 2019. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*. PMLR, 6105–6114.
[43] Run Wang, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Yihao Huang, Jian Wang, and Yang Liu. 2021. FakeSpotter: A simple yet robust baseline for spotting AI-synthesized fake faces. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*. 3444–3451.
[44] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. 2020. CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8695–8704.

[45] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1492–1500.
[46] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. 2019. Detecting and simulating artifacts in GAN fake images. In *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS'19)*. IEEE, 1–6.
[47] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 633–641.
[48] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2223–2232.

Received 20 April 2023; revised 16 November 2023; accepted 25 February 2024