Unveiling the Impact of Image Transformations on Deepfake Detection: An Experimental Analysis

Federico Cocchi[1,2], Lorenzo Baraldi[2](✉), Samuele Poppi[1,2],
Marcella Cornia[1], Lorenzo Baraldi[1], and Rita Cucchiara[1,3]

[1] University of Modena and Reggio Emilia, Modena, Italy
{federico.cocchi,samuele.poppi,marcella.cornia,lorenzo.baraldi,
rita.cucchiara}@unimore.it
[2] University of Pisa, Pisa, Italy
{federico.cocchi,samuele.poppi,lorenzo.baraldi}@phd.unipi.it
[3] IIT-CNR, Pisa, Italy

Abstract. With the recent explosion of interest in visual Generative AI, the field of deepfake detection has gained a lot of attention. In fact, deepfake detection might be the only measure to counter the potential proliferation of generated media in support of fake news and its consequences. While many of the available works limit the detection to a pure and direct classification of fake versus real, this does not translate well to a real-world scenario. Indeed, malevolent users can easily apply post-processing techniques to generated content, changing the underlying distribution of fake data. In this work, we provide an in-depth analysis of the robustness of a deepfake detection pipeline, considering different image augmentations, transformations, and other pre-processing steps. These transformations are only applied in the evaluation phase, thus simulating a practical situation in which the detector is not trained on all the possible augmentations that can be used by the attacker. In particular, we analyze the performance of a k-NN and a linear probe detector on the COCOFake dataset, using image features extracted from pre-trained models, like CLIP and DINO. Our results demonstrate that while the CLIP visual backbone outperforms DINO in deepfake detection with no augmentation, its performance varies significantly in presence of any transformation, favoring the robustness of DINO.

Keywords: Deepfake Detection · Self-Supervised Vision Transformers

1 Introduction

Although the generation of deepfake encompasses results of diverse nature, the world of fake image forgery has gained a lot of attention, since the breakthrough of diffusion models [7,13,30,31,33] in the Generative AI domain. While this technological advancement was received enthusiastically by the community, it has

F. Cocchi and L. Baraldi—Equal contribution.

G. L. Foresti et al. (Eds.): ICIAP 2023, LNCS 14234, pp. 345–356, 2023.
https://doi.org/10.1007/978-3-031-43153-1_29

also raised significant concerns regarding its potential impact on various domains, including the realms of human art and privacy. Both these domains are susceptible to risks due to the ease with which these models generate new content. Consequently, in light of the ongoing advancements in Generative AI, there has been a significant shift towards enhancing deepfake detection systems [36,41] to mitigate the risks posed by the remarkably convincing nature of such content.

The first efforts towards AI-generated content detection were conceived in the realm of fake face detection, with the release of ad-hoc datasets [18,32] and methodologies [11,19]. However, it should be noted that the significance of deepfake detection extends beyond fake faces or biometric data, necessitating the need for broader and more versatile detection methods that can address a wider range of generative scenarios. Only recently, a limited number of studies [1,6,36] have started to investigate deepfake images generated from text-to-image models [2,30,31,33], thereby enabling the detection of a wider variety of subjects with respect to biometric data. Although these studies assert high accuracy in detecting fake images, the resilience and robustness of the proposed methods have not yet been quantitatively evaluated.

In this manuscript, we freeze the recently proposed Stable Diffusion [31] model as the text-to-image generator and test two different detection approaches. In addition, we employ two different feature extractors, namely CLIP [29] and DINO [4], and evaluate their robustness to a wide variety of image transformations, at pixel-value and image-structure levels (Fig. 1). To the best of our knowledge, we are the first to assess the performance variability of real-fake recognition within such an environment. The experimental results shed light on the generally more robust performance of self-supervised methods (*i.e.*, DINO) against transformations in deepfake detection. Indeed, while CLIP achieves better performance without augmentation, the behavior of deepfake classifiers across different transformations is more consistent for DINO compared to CLIP. Surprisingly, CLIP performs similarly to DINO in the recognition of real images.

2 Related Work

Text-to-Image Generation. Deepfake images can be generated through three main models which consist of autoregressive approaches [25,26,33,39], generative adversarial networks (GANs) [12,34,38,42], and diffusion models [7,13,17,37]. In this work, we narrow down the field of deepfake generation considering the recent paradigm of text-to-image generation, which consists of generating an image starting from a textual description. While some GAN-based approaches [20] have been proposed as a possible solution to text-to-image generation, great results have been recently obtained with the application of diffusion models [2,30,33] by conditioning the diffusion process on the input textual description.

Recently, latent diffusion models [28,31] have improved the efficiency of standard diffusion models while maintaining their generation quality, by operating in a lower dimensional latent space z using a pre-trained variational autoencoder (VAE) [9,16]. In particular, during image generation, this approach involves the diffusion process occurring within the embedding space z, followed by the decompression of the resulting image through the VAE decoder. We conduct our experiments using images generated by the Stable Diffusion model [31], using both the 1.4 and 2.0 versions. The main differences between them lie in the backbone used to extract features from texts and images. In fact, Stable Diffusion v1 employs CLIP [31], which is trained on a non-publicly available dataset, while Stable Diffusion v2 relies on OpenCLIP [14], which is trained on a subset of LAION-5B [35] dataset. Both Stable Diffusion versions are finetuned on a filtered subset of LAION-5B to improve aesthetics and avoid explicit contents.

Deepfake Detection. The deepfake detection pipeline employed in this study comprises two consecutive stages: an image feature extractor followed by the actual detector. As for the first bit, different works have made extensive use of CLIP features as a starting point for their analysis [1,24,36]. In [5], they introduced an exploratory study of the frequency spectrum of the created images, thus capturing the impact of the specific generation model on the structure of the final images. Conversely, in [1], the authors proposed a wider-spectrum evaluation of the effects of different image feature extractors, presenting results on CLIP and OpenCLIP. Simultaneously, within the literature on image watermarking [10], analyses have been conducted to examine the robustness of the added watermark when the image is subjected to transformations. This type of analysis has been also conducted in relation to the detection of manipulated images and videos specifically focused on facial manipulation [22]. We embark on this path, applying it to the deepfake detection scenario, and studying how it affects the performance of some detection algorithms and the distribution of the features in the embedding space.

3 Evaluation Framework

3.1 Dataset

This section provides an overview of the COCOFake dataset [1] used in this work to perform the analysis on deepfake detection. COCOFake consists of an extension of the COCO dataset [21], that includes both real and fake images. Specifically, each real image in COCO is paired with five captions which are used to generate five fake images through a text-to-image model. The dataset is divided into training, validation, and test sets following the Karpathy splits, as used in the captioning literature [15]. Since COCO contains 113,287 training images and 5,000 validation and test images, COCOFake is composed of 679,722 instances in training, and 30,000 in validation and test.

From a technical standpoint, the production of counterfeit images is achieved through the utilization of Stable Diffusion [31] version 1.4. Furthermore, COCOFake also includes validation and test splits generated with Stable Diffusion version 2.0 to increase the robustness and generalization of possible analysis. It is worth mentioning that, all the images of COCOFake are stored in JPEG format, following the original COCO compression.

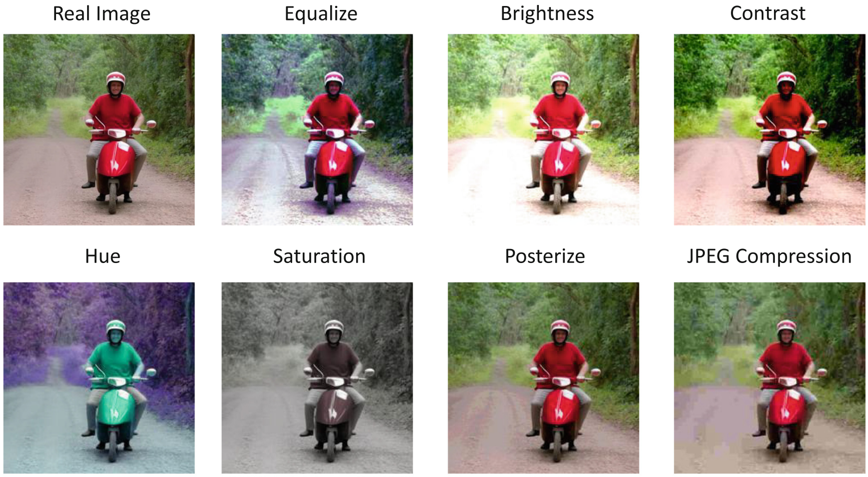

Fig. 1. Visual comparison of image transformations on a sample real image (top left).

3.2 Visual Backbones

In our experimental analysis, we employ three different visual backbones, namely CLIP [29], DINO [4], and DINOv2 [27]. It is worth mentioning that all the backbones adopt the same Vision Transformer architecture [8], ensuring a fair comparison between the employed methods.

The primary distinction among the visual backbones is the pre-training method employed. For instance, the CLIP approach utilizes language supervision to enforce similarities between visual and textual concepts. This is achieved by independently processing the image and its textual description using a visual and a textual backbone and then linearly projecting their representation into a shared embedding space. CLIP is pre-trained with a contrastive objective that maximizes the cosine similarity of correct image-text pairs. While CLIP obtains a semantic coherence [23] that can be useful for deepfake detection, the only image augmentation that is applied during training consists of a random square crop from resized images. This could make the visual backbone vulnerable to adversarial image augmentation.

In contrast to CLIP, DINO eschews the use of textual references, heavily relying on image augmentations during the pre-training phase. Indeed, DINO augments the input image through various techniques, including multi-crop [3], color jittering, Gaussian blur, and solarization. Multi-crop is used to generate multiple views of the same image, which can be logically divided into local views with lower resolutions and global views with higher resolutions. The DINO model is trained by enforcing local-to-global correspondences between different views of the same image. On the other hand, DINOv2 introduces additional pre-training objectives compared to DINO, such as randomly masking patches of the local views, leaving the model to learn how to reconstruct these patches.

Since both DINO and DINOv2 enforce robustness to image augmentation during pre-training, we investigate their effectiveness in a deepfake detection pipeline.

3.3 Deepfake Detection Pipeline

In this section, we present the deepfake detection pipeline that has been utilized for the analysis conducted in this study. Our pipeline encompasses a feature extraction phase followed by a detector model. Specifically, the detector model under investigation includes both a linear probe and a k-nearest neighbor (k-NN) classifier. The incorporation of different detector models serves the purpose of assessing distinct aspects. Specifically, the linear probe is engineered to identify any potential indications of the generation process within the feature space. Conversely, the k-nearest neighbor approach relies on the distance between existing features stored during training, thus allowing us to measure the similarity between real and fake content, in the embedding space.

Feature Extraction Process. From a technical perspective, the previously introduced visual backbones are employed as feature extraction models. Indeed, during the process of feature extraction, each image from the training, validation, and test sets of COCOFake undergoes processing by the visual backbones CLIP, DINO, and DINOv2. It is worth mentioning that no image augmentation is applied during the feature extraction phase.

Formally, each image $x \in \mathbb{R}^{C\times H\times W}$ is firstly split into a sequence of squared patches $\{x_i^p\}_{i=1}^N$ where C, H, W are respectively channel, height and width, while $x_i^p \in \mathbb{R}^{P^2C}$ is the i-th image patch of size $P \times P$. Consecutively, the sequence of image patches is linearly projected in the embedding dimensionality of the model D. At this step, a learnable classification token `[CLS]` $\in \mathbb{R}^D$ is concatenated to the input sequence. After L self-attention blocks the `[CLS]` token is saved as the representation of the image. In addition, and only for the CLIP model, the `[CLS]` token is linearly projected into the multi-modal embedding space.

Implementation-wise, the Base version of ViT [8] (*i.e.*, ViT-B) is used for CLIP, DINO, and DINOv2. In detail, ViT-B includes 85M learnable parameters, a 768 embedding dimensionality D, and $L = 12$ self-attention blocks. The considered input image size is $C = 3$, $H = 224$, $W = 224$, while the image patch size P is 14 for DINOv2 and 16 for CLIP and DINO. Regarding the pre-trained weights, the open-source ViT-B/16 version (*i.e.*, OpenCLIP [14]), pre-trained on the LAION-2B dataset [35], is used for CLIP, while the publicly available ViT-B/16 and ViT-B/14 are used for DINO and DINOv2, respectively.

Linear Probe. In the linear probe approach, we use the extracted features to train a logistic regressor. The goal of the method is to identify a signature, or imprint, in the extracted features that enable the linear model to distinguish between real and fake data. The logistic regressor is trained with an ℓ_2 objective, and the loss is weighted to account for the difference in the number of real and fake samples. Specifically, since the number of fake images in COCOFake is five times greater than the number of real images, the loss is weighted inversely

Table 1. Comprehensive summary of essential information regarding the applied transformations to assess the robustness of the different classifiers.

Transformation	Parameter	Range		Type	
		Min	Max	Pixel	Structure
Equalize	–	–	–	✓	✗
Center Crop	size	64	512	✗	✓
Resize	size	64	512	✗	✓
Random Crop	size	64	512	✗	✓
Brightness	brightness factor	0.5	2.0	✓	✗
Contrast	contrast factor	0.5	2.0	✓	✗
Hue	hue factor	−0.5	0.5	✓	✗
Saturation	saturation factor	0.1	3.0	✓	✗
Posterize	bits	1	8	✓	✗
Gaussian Blur	kernel size	3	15	✓	✗
JPEG Compression	quality	10	90	✓	✗
SD Compression	–	–	–	✓	✗

proportional to class frequencies. In addition, the LBFGS solver [40] is employed for training. Results are evaluated with accuracy scores over real and fake data.

k-nearest neighbor (k-NN). The classification task in the k-nearest neighbor approach is dependent on measuring distances within the visual feature space extracted by the utilized backbones. This implies that no further training is required. Hence, in the validation and test sets, the distances between each element and the features stored offline from the training split are calculated. The deepfake classification task is a supervised task, whereby the corresponding label (real or fake) is known for each feature embedding. So, the accuracy is determined by applying majority voting on the k-nearest features within the training feature space.

While the k-NN approach was originally proposed by [24] in a deepfake detection scenario, it presents notable limitations. Specifically, k-NN is highly sensitive to missing values or outliers, necessitating extensive coverage in the embedding space of the visual backbones by the training dataset. Moreover, as the dataset size increases, the computational cost of calculating distances between a new image and each existing one escalates significantly, ultimately compromising the algorithm performance. From an implementation perspective, we take into account the cosine similarity and the top-1 nearest neighbor to define the k-NN. Moreover, to manage the unbalanced COCOFake dataset, only a single pair of real and fake images are considered to compute the visual features in the training split, thus obtaining balanced real-fake images.

3.4 Image Augmentation

Drawing inspiration from [10,22], we explore the effectiveness of twelve distinct image augmentation techniques, detailed in Table 1. This series of transformations depict the potential manipulations of the image, considering image-structure and pixel-value transforms. As we can notice, each augmentation

Table 2. Accuracy performance on the COCOFake test set without any transformations for Stable Diffusion v1.4 and v2.0, using different classifiers and backbones.

Backbone	Stable Diffusion v1.4		Stable Diffusion v2.0	
	Linear	k-NN	Linear	k-NN
CLIP	99.6	96.7	99.3	94.9
DINO	96.9	91.3	90.5	87.8
DINOv2	96.6	89.0	95.7	84.6

involves a tunable parameter to control the degree of impact on images. We undertake a detailed analysis of these parameters to assess the robustness of the classification methods in response to the strength of the transformation. To this end, we select a range delimited by a minimum and maximum parameter for each augmentation, aiming to preserve the visual quality of the image in both cases, thus ensuring the preservation of visual consistency and usability. We assess the results by linearly partitioning the parameter range into five equally spaced segments. Following this process, we obtain five different image augmentation techniques for each transform with varying strengths. The utilization of these transformations evaluates the employed classifiers' accuracy in terms of resilience and generalization. A visual example of some of the image augmentation applied to an image is reported in Fig. 1.

In addition to the conventional augmentation methods, we introduce a novel technique called Stable Diffusion (SD) compression. This approach involves the projection of an image x into the latent space z of the Stable Diffusion model by utilizing the encoder of the autoencoder model [9] implemented within the Stable Diffusion framework. Following this projection, the image x is reconstructed using the decoder of the autoencoder. This augmentation technique is exclusively applied to real images to examine the biases of the detector concerning the lossy compression inherent in the generation of fake images.

4 Experimental Results

In this section, we analyze the results obtained by employing data augmentation on real and fake images, while testing different visual backbones.

Deepfake Detection of Plain Images. To evaluate the resilience of the aforementioned methods, a preliminary study is conducted to examine the performance of the detection pipeline without any applied transformations.

Based on the findings presented in Table 2, we can notice that the linear probe classifier exhibits a high classification accuracy, across all the backbones, with scores of 99.6%, 96.9%, and 96.6%, respectively with CLIP, DINO, and DINOv2, over the COCOFake test set generated with Stable Diffusion v1.4. These results validate the hypothesis that linear probes effectively identify the

Table 3. Comparison of accuracy performance on the COCOFake test set with transforms applied to fake images. The table shows results for linear and k-NN classifiers, for each backbones.

Transformation	CLIP		DINO		DINOv2	
	Linear	k-NN	Linear	k-NN	Linear	k-NN
Equalize	$11.9_{\pm0.0}$	$87.4_{\pm0.0}$	$94.1_{\pm0.0}$	$92.5_{\pm0.0}$	$89.8_{\pm0.0}$	$89.2_{\pm0.0}$
Center Crop	$28.3_{\pm26.4}$	$85.7_{\pm18.4}$	$85.3_{\pm15.0}$	$89.9_{\pm7.3}$	$87.7_{\pm15.7}$	$83.9_{\pm10.4}$
Resize	$83.0_{\pm8.7}$	$95.5_{\pm1.2}$	$94.3_{\pm5.2}$	$93.7_{\pm0.2}$	$93.8_{\pm3.2}$	$89.2_{\pm1.1}$
Random Crop	$31.8_{\pm25.4}$	$84.1_{\pm21.1}$	$85.8_{\pm14.0}$	$88.6_{\pm9.2}$	$87.5_{\pm16.4}$	$83.0_{\pm11.6}$
Brightness	$29.1_{\pm27.9}$	$88.5_{\pm7.9}$	$91.9_{\pm4.1}$	$93.3_{\pm0.3}$	$95.5_{\pm0.8}$	$89.8_{\pm0.1}$
Contrast	$31.4_{\pm26.2}$	$90.0_{\pm5.6}$	$94.8_{\pm3.8}$	$93.7_{\pm0.4}$	$95.7_{\pm0.9}$	$89.8_{\pm0.2}$
Hue	$74.2_{\pm2.6}$	$93.5_{\pm1.6}$	$93.3_{\pm3.1}$	$93.4_{\pm0.4}$	$95.3_{\pm0.5}$	$89.4_{\pm0.4}$
Saturation	$56.8_{\pm28.6}$	$92.1_{\pm6.1}$	$94.1_{\pm5.8}$	$93.3_{\pm0.7}$	$95.8_{\pm3.1}$	$89.7_{\pm0.4}$
Posterize	$29.5_{\pm37.7}$	$77.2_{\pm24.2}$	$89.9_{\pm6.9}$	$92.4_{\pm1.4}$	$86.1_{\pm11.0}$	$87.4_{\pm3.5}$
Gaussian Blur	$31.6_{\pm33.5}$	$95.8_{\pm0.7}$	$88.1_{\pm6.7}$	$92.9_{\pm0.5}$	$94.0_{\pm1.7}$	$89.3_{\pm0.3}$
JPEG Compression	$50.3_{\pm29.3}$	$95.1_{\pm3.0}$	$97.7_{\pm2.1}$	$93.8_{\pm0.4}$	$96.7_{\pm3.0}$	$89.5_{\pm0.9}$
Average	$41.6_{\pm24.6}$	$89.5_{\pm9.0}$	$91.8_{\pm6.7}$	$92.5_{\pm2.1}$	$92.3_{\pm5.6}$	$88.2_{\pm2.9}$

generator's imprint, embedded in the image features. Similar behavior is also highlighted by the k-NN approach, whose objective is not to specifically identify the imprinting trace. The observed performance strongly suggests that, in the backbones embedding space, fake images tend to exhibit proximity to one another and a similar phenomenon may hold true for real images. Specifically, k-NN performs with an accuracy of 96.7%, 91.3%, and 89%, over respectively CLIP, DINO, and DINOv2.

Moreover, the comparable performance observed on the COCOFake test set of Stable Diffusion 1.4 and Stable Diffusion v2.0 underscores the classifiers' capability to generalize beyond their initial training domain. As a result, further experiments will solely focus on the test set of Stable Diffusion 1.4. Building upon these initial results, subsequent experiments extend the analysis to explore the accuracy patterns when transformations are applied to fake and real images.

Fake Data Analysis. Presented in Table 3, we encounter a concise overview of the performance of the deepfake detection pipeline over transformed fake images. Evidently, the evaluation using the linear probe on the CLIP backbone demonstrates remarkably low performance. Specifically, CLIP achieves an average accuracy, among all the transformations, of only 41.6% for fake images, while DINO and DINOv2 demonstrate higher accuracy of 91.8% and 92.3%, respectively. Furthermore, the average standard deviation of CLIP, which amounts to 24.6%, highlights the substantial variability in performance across different transformations. This variability poses a significant threat to the overall robust-

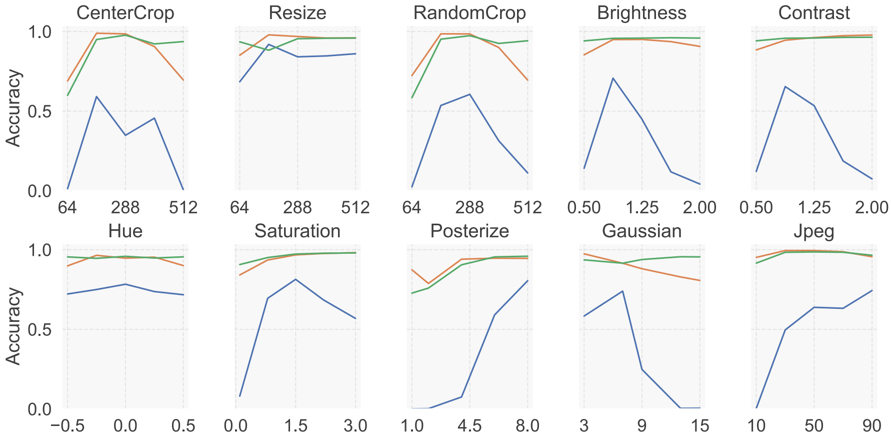

Fig. 2. The plots showcase the linear probe accuracy using different backbones, namely **CLIP**, **DINO**, and **DINOv2**, varying the applied transformation. Each subplot illustrates the accuracy of the classifiers under varying degrees of strength in image augmentations, used to provide insights into the effectiveness of classifiers.

ness of a CLIP-based deepfake detector. In contrast, DINO and DINOv2 consistently exhibit robustness across a wide range of performed transformations. In addition, Fig. 2 illustrates the trajectory of accuracy outcomes for the linear probes under varying degrees of strength of image augmentations, as discussed in Sect. 3.4. It is visually evident that, while DINO and DINOv2 exhibit a tendency to maintain consistent performance levels, CLIP performance is highly influenced by the intensity of each transformation. For example, a JPEG compression transformation with 10% quality produces an accuracy of 0.4% over CLIP linear probe while 95% and 91% for respectively DINO and DINOv2. We assume that the linear probe trained on CLIP-extracted features may be prone to overfitting on the distinctive imprint of fake data. This assumption arises from the observation that the CLIP visual backbone is not trained using extensive data augmentation. Consequently, alterations in the images could modify the extracted features, thus altering the fake imprint. This would explain the significant decline in the performance of the linear probe on CLIP. Although the k-NN outcomes, as shown in Table 3, indicate that CLIP achieves accuracy on par with DINO and DINOv2, the higher average standard deviation observed in CLIP highlights the superiority of the latter models.

Real Data Analysis. Table 4 presents a comprehensive analysis of the performance of CLIP, DINO, and DINOv2 evaluated on transformed real images. We decide to logically cluster results in `JPEG Compression`, `SD Compression`, and `Other Transforms` to facilitate the analysis. Specifically, we isolate the compression augmentations, leaving a summary of the others. Regarding the obtained results, it is noteworthy that the linear probes demonstrate commendable per-

Table 4. Accuracy performance on the COCOFake test set with transformations applied to real images. We report results for linear probe and k-NN classifiers for each backbones. Transformations are divided into compression based and others, to highlight the accuracy drop when applying compression-based transformations.

Backbone	JPEG Compression		SD Compression		Other Transforms	
	Mean	Std	Mean	Std	Mean	Std
Linear						
CLIP	93.2	12.5	44.2	–	99.7	0.5
DINO	74.8	15.2	80.1	–	93.2	6.0
DINOv2	58.2	20.4	54.2	–	91.9	5.5
k-NN						
CLIP	87.5	2.7	80.2	–	90.0	5.7
DINO	75.0	3.8	75.2	–	76.0	5.0
DINOv2	80.3	1.8	79.4	–	81.7	2.8

formance on the other non-compression-based transforms. However, when subjected to JPEG compression, the linear probes exhibit lower accuracy. Specifically, the average accuracy reaches 93.2%, 74.8%, and 58.2% for CLIP, DINO, and DINOv2 respectively. Furthermore, the poorest performance is observed in CLIP with SD compression, resulting in an accuracy of 44.2%. We hypothesize that the compression imprints bear a strong resemblance to the fake imprint, thereby deceiving the linear probe into misclassifying a real image as fake.

A comparable examination can be directly carried out on the feature space of the visual backbones. Specifically, when considering the embedding space of CLIP, real images subjected to the SD compression exhibit closer proximity, on average, to fake images compared to JPEG compression and other transformations. This is additional proof that SD compression has a great influence on the fake data imprint. In contrast, DINO and DINOv2 are equally subjected to all transformations, exhibiting an average accuracy in the k-NN analysis of 75.4% and 80.5%, respectively. It is noteworthy that the limitations inherent to k-NN, as mentioned in Sect. 3.3, can attenuate its impact on deepfake detection.

5 Conclusion

In conclusion, the growing capacity and utilization of text-to-image models present a persistent challenge in the detection of artificially generated images. Our proposal introduces an analysis of the robustness of a set of classifiers, specifically considering transformations that modify the visual appearance of the image. The performance of the classifiers is significantly influenced by these transformations and this study emphasizes the significance of the robustness to such transformations for deepfake detector classifiers that need to operate in real-world scenarios.

Acknowledgments. This work has partially been supported by the European Commission under the PNRR-M4C2 (PE00000013) project “FAIR - Future Artificial Intelligence Research” and by the Horizon Europe project “European Lighthouse on Safe and Secure AI (ELSA)” (HORIZON-CL4-2021-HUMAN-01-03), co-funded by the European Union (GA 101070617).

References

1. Amoroso, R., Morelli, D., Cornia, M., Baraldi, L., Del Bimbo, A., Cucchiara, R.: Parents and children: distinguishing multimodal DeepFakes from natural images. arXiv preprint arXiv:2304.00500 (2023)
2. Balaji, Y., et al.: eDiff-I: text-to-image diffusion models with an ensemble of expert denoisers. arXiv preprint arXiv:2211.01324 (2022)
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: NeurIPS (2020)
4. Caron, M., et al.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
5. Corvi, R., Cozzolino, D., Poggi, G., Nagano, K., Verdoliva, L.: Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In: CVPR Workshops (2023)
6. Corvi, R., Cozzolino, D., Zingarini, G., Poggi, G., Nagano, K., Verdoliva, L.: On the detection of synthetic images generated by diffusion models. In: ICASSP (2023)
7. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. In: NeurIPS (2021)
8. Dosovitskiy, A., et al.: An image is worth 16 × 16 words: transformers for image recognition at scale. In: ICLR (2021)
9. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021)
10. Fernandez, P., Sablayrolles, A., Furon, T., Jégou, H., Douze, M.: Watermarking images in self-supervised latent spaces. In: ICASSP (2022)
11. Ganguly, S., Ganguly, A., Mohiuddin, S., Malakar, S., Sarkar, R.: ViXNet: Vision Transformer with Xception Network for DeepFakes based video and image forgery detection. Expert Syst. Appl. **210**, 118423 (2022)
12. Goodfellow, I., et al.: Generative adversarial nets. In: NeurIPS (2014)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
14. Ilharco, G., et al.: OpenCLIP (2021). https://doi.org/10.5281/zenodo.5143773
15. Karpathy, A., Li, F.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
16. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 (2013)
17. Kingma, D.P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., Welling, M.: Improved variational inference with inverse autoregressive flow. In: NeurIPS (2016)
18. Li, L., Bao, J., Yang, H., Chen, D., Wen, F.: Advancing high fidelity identity swapping for forgery detection. In: CVPR (2020)
19. Li, L., et al.: Face X-Ray for more general face forgery detection. In: CVPR (2020)
20. Liao, W., Hu, K., Yang, M.Y., Rosenhahn, B.: Text to image generation with semantic-spatial aware GAN. In: CVPR (2022)

21. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
22. Lu, Y., Ebrahimi, T.: Assessment framework for DeepFake detection in real-world situations. arXiv preprint arXiv:2304.06125 (2023)
23. Mukhoti, J., et al.: Open vocabulary semantic segmentation with patch aligned contrastive learning. In: CVPR (2023)
24. Ojha, U., Li, Y., Lee, Y.J.: Towards universal fake image detectors that generalize across generative models. In: CVPR (2023)
25. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with PixelCNN decoders. In: NeurIPS (2016)
26. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
27. Oquab, M., et al.: DINOv2: learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
28. Peebles, W., Xie, S.: Scalable diffusion models with transformers. arXiv preprint arXiv:2212.09748 (2022)
29. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
30. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125 (2022)
31. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
32. Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: Face-Forensics++: learning to detect manipulated facial images. In: ICCV (2019)
33. Saharia, C., et al.: Photorealistic text-to-image diffusion models with deep language understanding. In: NeurIPS (2022)
34. Sauer, A., Karras, T., Laine, S., Geiger, A., Aila, T.: StyleGAN-T: unlocking the power of GANs for fast large-scale text-to-image synthesis. arXiv preprint arXiv:2301.09515 (2023)
35. Schuhmann, C., et al.: LAION-5B: an open large-scale dataset for training next generation image-text models. In: NeurIPS (2022)
36. Sha, Z., Li, Z., Yu, N., Zhang, Y.: DE-FAKE: detection and attribution of fake images generated by text-to-image diffusion models. arXiv preprint arXiv:2210.06998 (2022)
37. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)
38. Tao, M., Bao, B.K., Tang, H., Xu, C.: GALIP: generative adversarial CLIPs for text-to-image synthesis. In: CVPR (2023)
39. Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: ICML (2016)
40. Xiao, Y., Wei, Z., Wang, Z.: A limited memory BFGS-type method for large-scale unconstrained optimization. Comput. Math. Appl. **56**(4), 1001–1009 (2008)
41. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: rooting DeepFake attribution in training data. In: ICCV (2021)
42. Zhang, H., et al.: StackGAN: text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV (2017)