Leveraging High-Frequency Components for Deepfake Detection

Nesryne Mejri, Konstantinos Papadopoulos, Djamila Aouada
Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg
Luxembourg
nesryne.mejri@ext.uni.lu, papad.konst@gmail.com, djamila.aouada@uni.lu

***Abstract*—In the past years, RGB-based deepfake detection has shown notable progress thanks to the development of effective deep neural networks. However, the performance of deepfake detectors remains primarily dependent on the quality of the forged content and the level of artifacts introduced by the forgery method. To detect these artifacts, it is often necessary to separate and analyze the frequency components of an image. In this context, we propose to utilize the high-frequency components of color images by introducing an end-to-end trainable module that (a) extracts features from high-frequency components and (b) fuses them with the features of the RGB input. The module not only exploits the high-frequency anomalies present in manipulated images but also can be used with most RGB-based deepfake detectors. Experimental results show that the proposed approach boosts the performance of state-of-the-art networks, such as XceptionNet and EfficientNet, on a challenging deepfake dataset.**

I. Introduction

Deepfakes are images and videos that seem genuine to the human eyes, whereas, in reality, they are either entirely or partially generated by an artificial intelligence algorithm. Deepfakes appeared in 2017 as adult forged content, depicting faces that were swapped with celebrities' faces [1]. As a technology, deepfakes have creative applications in movie post-production, dubbing, productive education, and identity anonymization. Nevertheless, they remain a significant threat to the public order and international peace[1], especially with the virality of social media [2].Consequently, developing automated deepfake detection tools has become a pressing matter. *Deepfake detection* started growing alongside the development of deepfake generation methods and open-source software like Face-Swap[3], FakeApp[4] and DeepFaceLab [6]. Researchers started building image and video databases of fake content, focusing on key properties, such as visual quality, level of artifacts, and setup diversity. Most of deepfake detectors operate on RGB data, as it is the most abundant form [5]. These detectors can be artifact-specific or undirected [7]. In the first case, they try to find particular anomalies produced by the deepfake generation methods. Such irregularities can manifest as inconsistencies in the noise level, the color, the spectrum in the frequency domain, or the time domain in the case of videos. On the other hand, undirected detectors are networks that decide on their own which features are the most relevant for classification between real and fake samples.

Promising results have been achieved so far [5]; however, most approaches are either too artifact-specific or too general, which hinders their performance. Indeed, methods extracting only one type of artifacts can perform well, given samples that contain these particular anomalies; nevertheless, they become unusable when presented with data that does not suffer from the targeted artifacts. For this matter, several works [9, 25, 34, 36, 37] adopted a mixture of color and frequency-domain related artifacts extracted in parallel, which seemed to be more sustainable. However, implementing those approaches can generally imply different actions. For example, it can be required to collect additional training data, especially when the input videos are heavily compressed [13]. Another requirement can be revisiting the architecture of a CNN [36], or tailoring a specific architecture [9]. Clearly, mixing color and frequency-domain artifacts is a suitable option for a more robust and accurate detection. Nevertheless, the complexity and lack of flexibility of the models remain unsettled. This problem is addressed within this work by presenting a simple and lightweight face-swap deepfake detection framework.

In this context, we propose to find a balance between artifact-specific and undirected approaches. We aim to improve the performance and the convergence of deepfake image-based detectors by guiding them towards finding anomalous high-frequency features in fake frames. Our approach does not rely on collecting any additional data; instead, it leverages information already present in the RGB input. We take interest in the noise present in the high-frequency components to discriminate between real and fake faces. Indeed, noise easily affects the high frequencies of signals, and we hypothesize that deepfake generation methods produce high-frequency noise whose distribution is dissimilar to the noise in the real image.

This paper proposes to leverage the high-frequency components of color images by extracting high-frequency and RGB features in parallel. Then, both types of features are fused and fed to a backbone network for classification. The advantage of this approach is that it only substitutes the first layer of a CNN, which makes it transferable to any RGB-based deepfake detector. It takes advantage of color information in

[1]https://www.theguardian.com/world/2021/apr/22/european-mps-targeted-by-deepfake-video-calls-imitating-russian-opposition
[2]https://www.bbc.com/news/technology-49961089
[3]https://github.com/deepfakes/faceswap
[4]https://www.malavida.com/en/soft/fakeapp

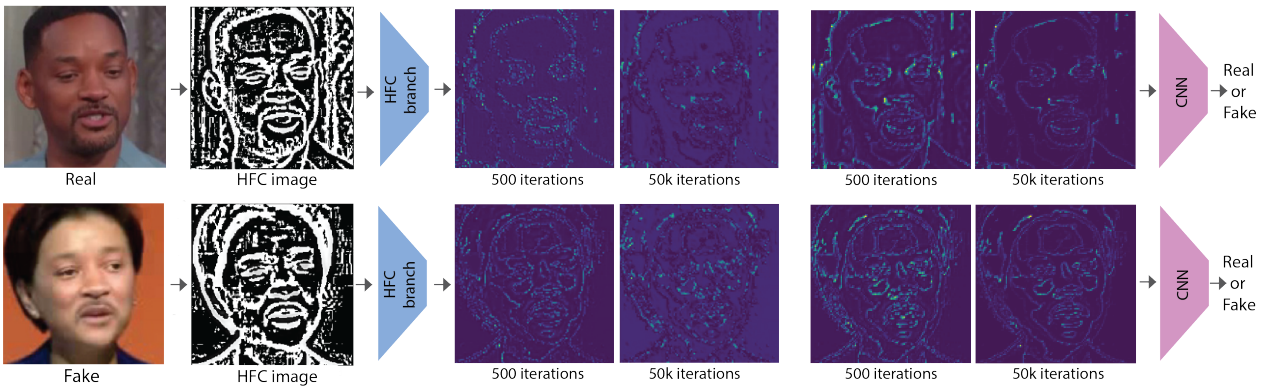

Fig. 1. An example of a pair of activation maps learned in the High-Frequency Components (HFC) branch. The features are extracted from the same face, and shown for different numbers of iterations. Real images do not show many noise traces even after a large number of iterations, whereas the fake images have their high-frequency components emphasized especially in the facial region.

an undirected manner but guides the detection with the noise traces. Additionally, it does not add an impactful cost to the overall network architecture. Fig. 1. shows an overview of the learned high-frequency traces by the proposed approach.

In summary, this paper's contributions are the following:

- a DNN framework that boosts the performance of detectors by leveraging the high-frequency components of color images.
- an easily adaptable module that extracts high-frequency components and fuses them with RGB features without inducing any domain discrepancy between the two types of features.
- an experimental validation and extensive analysis of the impact of the high-frequency components module, as well as the impact of identity bias on the performance of deepfake detectors.

This paper is organized as follows: Section II introduces a brief review of the literature of RGB-based and frequency domain-based deepfake detection. In Section III, a description of the proposed approach and its modalities is presented. The conducted experiments are given in Section IV. Finally, Section VI concludes this work.

II. Related Work

This section examines important works addressing deepfake detection using color images and approaches leveraging frequency-domain features for the same task.

*a) **RGB image-based deepfake detection**:* RGB-image-based detectors exist for both artifact-specific and undirected deepfake detectors [7]. The artifact-specific case is the most common one [8, 16, 18]–[21, 24, 26]–[28]. The work in [16] uses speeded up robust features (SURF) [17] and support vector machines (SVM) to discriminate between real and swapped faces. MesoNet [8], a shallow CNN network that extracts steganalysis and mesoscopic features, classifies videos based on an aggregation score made on each frame. Similarly, [18] detects fake videos by investigating frames for warping and blurring artifacts, as the resolution of generated faces is usually lower than final frame's resolution. Face X-ray [19] predicts the blended boundaries on face-swap deepfakes. [20, 21] demonstrated that GAN-generated content hides a unique fingerprint that can be attributed to the generation method. Approaches such as [24] focus on prepossessing the input images with a specific module that emphasizes the residual artifacts, then using adaptive convolutional layers, learns to recognize those particular artifacts. Physiological artifacts have also been explored; [26]–[28] use RGB frame sequences to measure the heart rate and the pulse of subjects in videos from their skin tone.

In the undirected case, XceptionNet [4], a CNN network, achieved encouraging results on different face-swap forgery methods [13]. Besides, achieving over 95% accuracy on raw images, its performance with respect to different compression rates remains competitive compared to other methods. Similarly, the winner of the Facebook Deepfake Detection Challenge [15, 29] and [30] used a state-of-the-art EfficientNet [3] as a backbone network which proved to be effective in deepfake classification. These works showed that undirected methods are as efficient as artifact-specific approaches. The key factors in building a powerful model are the backbone selection and well-crafted data augmentation. Generally, color-image-based detectors are performant. However, they either require a significant amount of data to perform well [13], or they only target one type of artifacts, which puts them at risk when the artifacts are not present in the images.

*b) **Frequency-domain deepfake detection**:* Besides color information, many works consider frequency-domain features for deepfake detection [22, 23, 31]–[33, 35]. Wang et al. showed in [22] that the high-frequency components play a significant role in the generalization capabilities of CNNs, whereas [31] proved that learning in the frequency-domain could preserve most of the information within high-resolution images. [23] pointed that using the Discrete Fourier Transform (DFT) and averaging the amplitude of each frequency band can reveal discriminative spectral irregularities in fake faces. F3-Net [32] used two frequency-domain-based pipelines: One

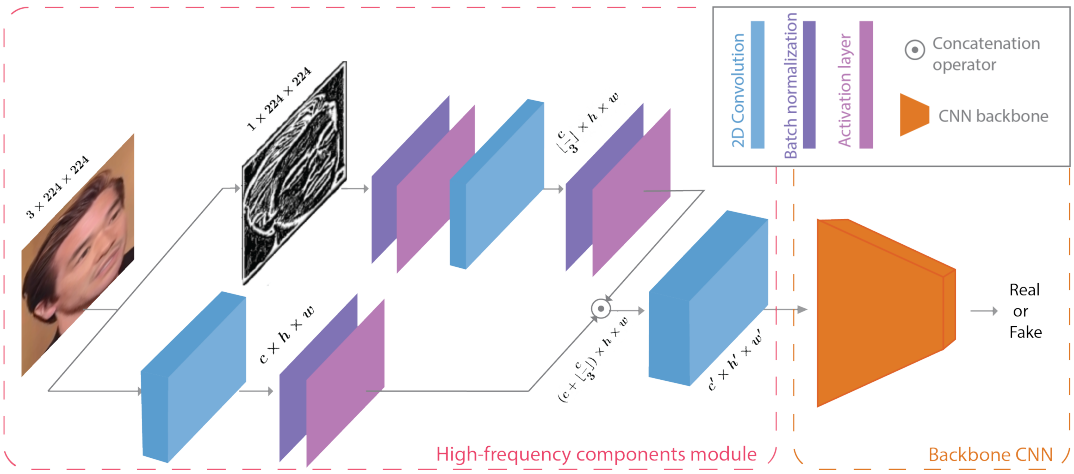

Fig. 2. A deepfake detection architecture augmented with High-Frequency Components (HFC) module. On the left, the module substitutes the first layer of a CNN network. It preprocesses the RGB input image to extract high frequency components as well as color features in parallel. On the right is the CNN backbone which can be any state-of-the-art network like XceptionNet [4] or EfficientNet [3].

that discovers salient frequencies from the Discrete Cosine Transform (DCT) and the second extracting local frequency statistics. [33] showed that GAN-generated content exhibits severe spectral artifacts stemming from the upsampling step of generative models and their variants. [35] showed that CNN-generated content has traceable artifacts common to all forgery methods and that by performing adequate data augmentation, models generalize well to both unseen tampering techniques and datasets. Similar to color-image-based detection, using only one type of artifact is not always reliable for detection.

*c) **Mixed artifacts deepfake detection**:* The third direction explored for deepfake detection uses both color and frequency-domain features [25, 34, 36, 37]. The works proposed in [25, 36, 37] rely on two streams, where color features are learned in combination with frequency-domain-related features. Similarly, [34] presents a new approach (SPSL) that uses the spatial images and the phase spectrum jointly to capture upsampling artifacts. Furthermore, [36] shows that most deepfake detection models are biased on method-specific color textures and that high-frequency noise features yield more robust representations, which reduces overfitting.

In general, targeting more than one type of artifacts improves the detectors' performance. However, it usually comes at the cost of developing complex architectures. Therefore, we propose to build richer feature representations by combining high-frequency components' noise with color features and keeping the overall architecture simple.

III. Proposed Approach

This section describes the proposed approach for RGB image-based face forgery detection, i.e., given an image or a video frame, detecting whether faces are real or forged. The solution consists of two independent parts: (1) a module for extracting different artifact-specific features jointly, followed by (2) a CNN backbone for classification. The introduced architecture is depicted in Fig. 2.

A. High-Frequency Components module

It is usually assumed that CNNs implicitly extract high-frequency components on their own as low-level features, since such information is always present within the edges of an image. However, we show that explicitly providing networks with these components helps them achieve a boost of performance at a minimal cost. Fig. 2., shows the proposed High-Frequency Components (HFC) module, which is transferable from one CNN network to another. It has two branches; The first is for exploiting the color features and has similar parameters as the first convolutional layer of the chosen backbone network. The convolutional layer of the color branch has an output of dimension $\mathcal{C}$ for the feature maps. Its goal is to extract low-level features from the color texture. The second branch is for exploiting high-frequency components. The high-frequency features are extracted in the image domain by first converting the input to a grayscale image I_g. Then, I_g is smoothed by a Gaussian filter. Finally, both I_g and its smoothed version are used to calculate the high-frequency image I_{HFC} as follows:

$$I_{HFC}(x,y) = I_g(x,y) - \frac{1}{2\pi\sigma^2}\sum_{i,j} I_g(x+i, y+j) e^{-\frac{i^2+j^2}{2\sigma^2}}, \tag{1}$$

where (x,y) are a given pixel coordinates, and σ the standard deviation of the Gaussian filter whose value is set empirically. The high-frequency image undergoes batch normalization and an activation layer, to be finally passed to a convolutional layer for high-frequency noise feature extraction. The latter layer outputs $\lfloor \frac{\mathcal{C}}{3} \rfloor$ feature maps, where $\lfloor \rfloor$ is the rounding operation. The extracted features are normalized and passed through an activation layer. Finally, the color features and the high-frequency features are stacked to obtain an output of dimension $\mathcal{C} + \lfloor \frac{\mathcal{C}}{3} \rfloor$ feature maps. The convolutional layer that accepts these feature maps outputs $\mathcal{C}'$ channels where $\mathcal{C}'$ matches the input dimension required by the second layer of

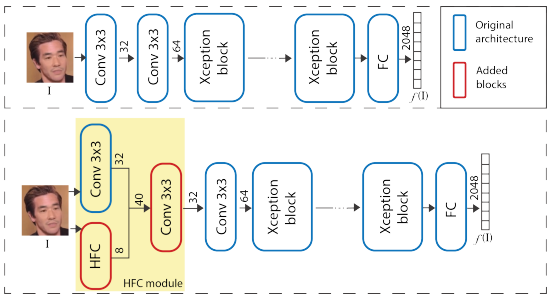

Fig. 3. The original XceptionNet architecture (top) versus the HFC augmented Xception architecture (bottom).

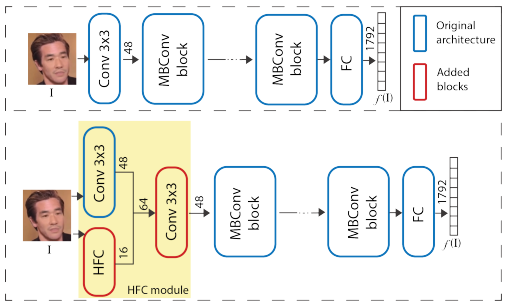

Fig. 4. The original EfficientNet architecture (top) versus the HFC augmented EfficientNet architecture (bottom).

the chosen backbone network.

B. Backbone networks

This subsection introduces the second main component of the proposed approach, i.e., CNN backbones compatible with the afore mentioned HFC module for deepfake detection.

a) ***XceptionNet****:* is a CNN trained and tested on ImageNet [42], on which it achieved 79% top-1 accuracy. It is characterized by its Xception block, inspired from the Inception block [2]. It differs from the latter by the performing depth-wise separable convolutions with residual connections. Each channel of the block's input is convolved separately, then a point wise convolution is applied. Its goal is to help the network capturing cross-channel correlations. It had been first proposed for face forgery detection in [13], and quickly became a baseline model for many works [43]–[47] performing the same task.

b) ***EfficientNet****:* is a more recent family of CNNs proposed in [3]. It introduced a compound CNN scaling approach. Its main idea is that convolutional neural networks cannot be randomly scaled and that their width, depth, and resolution depend on each other. Thus scaling CNNs, should be done uniformly according to a fixed ratio, i.e., when an input image has a larger resolution. In that case, the network needs more convolutional layers to increase its receptive fields and more channels to capture more fine-grained patterns on the bigger image. EfficientNet was trained and tested on ImageNet [42], on which it reached 83.8 % top-1 accuracy. It became popular due to the fair trade-off it offers in terms of dimension, complexity and classification performance [29, 30, 43, 47]. For our approach, we adopt Efficient-Net B4 as it has a relatively low number of parameters (19M), and because it proved to be efficient for face forgery detection [30].

IV. Experiments

This section outlines the setup and the led experiments.

A. Dataset

For our experiments, we chose to use Celeb-DF [14], a challenging state-of-the-art face-swapping videos dataset. It depicts 59 celebrities and provides a large number of fake videos where resembling identities are swapped between each other. The videos are generated by a learning-based method called *Deepfakes*[5], that had also been used to generate previous datasets such as FaceForensics++ [13]. However, Celeb-DF [14] is more challenging because its generation method performs a lot of post-processing to improve not only the visual quality but also the resolution of the generated faces. It also considers diversity on different levels (background, illumination, gender, skin color, age). The enhanced method generates 5,639 fake videos from a set of celebrities' videos downloaded from YouTube. On the other hand, there are only 590 videos in the real class, and not all of the subjects are celebrities. Therefore, the classes are imbalanced. Additionally, the sequences have an average length of 13 seconds, and most of the subjects are nearly front-facing without occlusions. Finally, all videos are in MPEG4.0 format to mimic a real-world setup. Table I provides Celeb-DF's [14] specifications.

TABLE I
SPECIFICATIONS OF THE CELEB-DF DATASET.

	#videos		#frames		#subjects	
Dataset	Real	Fake	Real	Fake	famous	regular
Celeb-DF [14]	590	5639	225.4k	2116.8k	59	300

B. Setup

We propose to split our dataset according to the cross-subject protocol. The models are trained only with 32 frames per sequence. Indeed, a smaller number would result in overfitting, whereas a larger one would not necessarily improve the model's performance [29, 30]. For each frame, the face tracking module BlazeFace [38] is employed to focus solely on the facial region. The extracted faces are saved as RGB images of size 224x224. The Albumentations [39] library is also used to augment the dataset, adding various transformations like random horizontal flipping, random hue, saturation, brightness, and contrast changes. The Gaussian filter size and σ are set empirically to 3×3 and $\sigma = 3$ respectively. The models are trained in an end-to-end fashion using the Pytorch [40] framework. We use the Binary Cross Entropy (BCE) loss and the Adam [41] optimizer with default parameters of $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$, as well as an initial learning rate of 10^{-3}. A decay factor of 0.1 is set to decay the learning rate as soon as the loss does not decrease after ten validation

[5]https://github.com/deepfakes/faceswap

passes. The batch size is 32 images taken in a random but balanced manner. Each model is trained up to 100k iterations. The training is set up to stop if the loss reaches a plateau after ten validation passes or if the minimum learning rate of 10^{-8} is reached. All the models are trained using a computer with an Intel Xeon E5-2640-v4 CPU and an NVIDIA Titan V GPU.

V. Results

This section reports the experimental results, highlighting the importance of proper data splitting protocols and the impact of early coupling of the high-frequency components with color data deepfake detection.

A. Analysis of the impact of the identity bias

One of the reasons deepfake detectors perform well on in-house deepfake generation methods and datasets is that they overfit the color textures of the input images as pointed out by [36]. In the same context, we show that detectors are also biased to the subjects' identities. Indeed, face-swap methods rely on gathering a large number of videos with real subjects and swapping the faces of those who are similar. Most deepfake detectors do not consider that when performing data splits. For this matter, we define the *identity split* as splitting the dataset according to the number of identities in it, instead of the number of videos, i.e., we put real and fake videos of the same person in the same set. We perform an identity split on Celeb-DF [14], using about 21% of the identities for testing and 19% for validation. Table II shows the impact of isolating the subjects' identities versus using a random data split proposed by the authors of [14]. The results suggest that the models' performance is biased and that they overfit the subjects' identities. Indeed, detectors learn the subjects' faces during training, which explains their high-performance on a testing set containing faces they know.

TABLE II
THE IDENTITY BIAS INDUCED BY THE RANDOM DATA SPLITS PROTOCOL.

Method	Accuracy	AUC
XceptionNet [4] + random split	0.9633	0.9951
XceptionNet [4] + *identity split*	0.8834	0.9669
EfficientNet-B4 [3] + random split	0.9642	0.9959
EfficientNet-B4 [3] + *identity split*	0.9056	0.9756

B. Analysis of the impact of HFC

Besides isolating subject identities when performing the data split, it is possible to improve the detection by making the model focus on more than one type of artifacts. We experiment with both XceptionNet [4], EfficientNet [3] and the HFC module. Table III shows the results of using our module with both networks. Despite their initial accurate performance, the HFC module could still improve the accuracy without using any additional data or changing the architecture of the backbone networks. The advantage of our method is that it allows to remain architecturally flexible and at the same time guide the supposedly undirected CNN with extracted high-frequency components, which boosts the performance and speeds up the network convergence. Additionally, we trained the backbone models with the high-frequency components of each channel, extracted as in (1) and stacked as an RGB image. The results suggest that high-frequency components provide complementary information to the low-frequency components and color features. Using such information explicitly leads to richer feature representations, making models less prone to overfitting the artifacts of the input images. Furthermore, from table IV, EfficientNet-HFC converges 12.5% faster at a cost of 28.65% additional FLOPS. On the other hand, XceptionNet-HFC benefits from a 48.15% faster convergence at the cost of 1.37% additional FLOPS. Thus, the results align with our goal of improving the performance and keeping the complexity low.

TABLE III
THE PERFORMANCE OF SOTA NETWORKS ON FACE FORGERY DETECTION WITH AND WITHOUT THE HIGH-FREQUENCY COMPONENTS FOLLOWING DIFFERENT DATA SPLITTING PROTOCOLS.

		Random split		Cross-id split	
Method	Setup	Acc.	AUC	Acc.	AUC
XceptionNet [4]	no HFC	0.9633	0.9951	0.8834	0.9669
	HFC	**0.9663**	**0.9963**	**0.9064**	**0.9728**
	HFC only	0.9431	0.9901	0.8750	0.9665
EfficientNet-B4 [3]	no HFC	0.9642	**0.9959**	0.9056	0.9756
	HFC	**0.9654**	0.9950	**0.9246**	**0.9782**
	HFC only	0.9198	0.9742	0.8123	0.9171

TABLE IV
MODELS PARAMETERS AND COMPLEXITY WITHIN THE PROPOSED FRAMEWORK

Method	#Param.	#FLOPS	#iter. to convergence
EfficientNet [3]	17.55 M	1.594 B	40.5k
EfficientNet [3] + HFC	17.58 M	2.234 B	36k
XceptionNet [4]	20.81 M	4.609 B	40k
XceptionNet [4] + HFC	20.82 M	4.673 B	27k

Additionally, Fig. 1 shows the activation maps of the high-frequency branch in the HFC module. As the number of iterations increases, the network learns to discard non-informative edges and only keeps the noise traces as the most informative regions of the images, especially for fake images. This experiment shows that the network is learning traces that are useful for deepfake detection.

VI. Conclusion

In this paper, we tackled the problem of face forgery detection by face-swapping methods. We showed that standard CNNs are capable of achieving accurate detection. However, their performance can suffer from identity bias when the data split is not performed carefully. Additionally, we introduced a flexible framework that uses state-of-the-art CNNs and high-frequency components to detect face tampering. We proposed letting the networks learn from the color features and, at the same time, guiding them towards finding noise traces in the high-frequency components of the input images. Contrary to other works leveraging the same type of features, our module is easily adaptable and can be transferred from one network to another without inducing heavy changes to the

overall architecture. Finally, the comprehensive experiments demonstrated the high-frequency features' informativeness and their role, alongside the color features, in building richer feature representations. It showed that our HFC module, and the identity split, can contribute to making deepfake detection models more precise. Future works will consider more challenging datasets with noise addition and different compression levels to simulate a more realistic environment.

REFERENCES

[1] Paris, Britt, and Donovan, Joan. "Deepfakes and Cheap Fakes." Data & Society, September 18, 2019.

[2] C. Szegedy et al., "Going deeper with convolutions," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[3] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional Neural Networks," arXiv [cs.LG], 2019.

[4] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[5] F. Juefei-Xu, R. Wang, Y. Huang, Q. Guo, L. Ma, and Y. Liu, "Countering malicious DeepFakes: Survey, battleground, and horizon," arXiv [cs.CV], 2021.

[6] I. Perov et al., "DeepFaceLab: A simple, flexible and extensible face swapping framework," arXiv [cs.CV], 2020.

[7] Y. Mirsky and W. Lee, "The creation and detection of deepfakes: A survey," ACM Comput. Surv., vol. 54, no. 1, pp. 1–41, 2021.

[8] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "MesoNet: A compact facial video forgery detection network," in 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 2018.

[9] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection," in 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017.

[10] X. Zhang, S. Karaman, and S.-F. Chang, "Detecting and simulating artifacts in GAN fake images," in 2019 IEEE International Workshop on Information Forensics and Security (WIFS), 2019.

[11] R. Tolosana, S. Romero-Tapiador, J. Fierrez, and R. Vera-Rodriguez, "DeepFakes evolution: Analysis of facial regions and fake detection performance," in Pattern Recognition. ICPR International Workshops and Challenges, Cham: Springer International Publishing, 2021, pp. 442–456.

[12] M. Du, S. Pentyala, Y. Li, and X. Hu, "Towards Generalizable Deepfake Detection with Locality-aware AutoEncoder," in Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020.

[13] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to detect manipulated facial images," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[14] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for DeepFake forensics," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[15] B. Dolhansky et al., "The DeepFake Detection Challenge (DFDC) Dataset," arXiv [cs.CV], 2020.

[16] Y. Zhang, L. Zheng, and V. L. L. Thing, "Automated face swapping and its detection," in 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), 2017.

[17] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in Computer Vision – ECCV 2006, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[18] Y. Li and S. Lyu, "Exposing DeepFake videos by detecting face warping artifacts," arXiv [cs.CV], 2018.

[19] L. Li et al., "Face X-ray for more general face forgery detection," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

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

[21] N. Yu, L. Davis, and M. Fritz, "Attributing fake images to GANs: Learning and analyzing GAN fingerprints," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[22] H. Wang, X. Wu, Z. Huang, and E. P. Xing, "High-frequency component helps explain the generalization of convolutional neural networks," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[23] R. Durall, M. Keuper, F.-J. Pfreundt, and J. Keuper, "Unmasking DeepFakes with simple Features," arXiv [cs.LG], 2019.

[24] Z. Guo, G. Yang, J. Chen, and X. Sun, "Fake face detection via adaptive manipulation traces extraction network," Comput. Vis. Image Underst., vol. 204, no. 103170, p. 103170, 2021.

[25] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[26] V. Conotter, E. Bodnari, G. Boato, and H. Farid, "Physiologically-based detection of computer generated faces in video," in 2014 IEEE International Conference on Image Processing (ICIP), 2014.

[27] U. A. Ciftci, I. Demir, and L. Yin, "FakeCatcher: Detection of synthetic portrait videos using biological signals," IEEE Trans. Pattern Anal. Mach. Intell., vol. PP, pp. 1–1, 2020.

[28] U. A. Ciftci, I. Demir, and L. Yin, "How do the hearts of deep fakes beat? Deep fake source detection via interpreting residuals with biological signals," in 2020 IEEE International Joint Conference on Biometrics (IJCB), 2020.

[29] Selim Seferbekov. https://github.com/selimsef/dfdc_deepfake_challenge.

[30] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of CNNs," arXiv [cs.CV], 2020.

[31] K. Xu, M. Qin, F. Sun, Y. Wang, Y.-K. Chen, and F. Ren, "Learning in the frequency domain," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[32] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in Computer Vision – ECCV 2020, Cham: Springer International Publishing, 2020, pp. 86–103.

[33] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," arXiv [cs.CV], 2020.

[34] H. Liu et al., "Spatial-Phase Shallow Learning: Rethinking face forgery detection in frequency domain," arXiv [cs.CV], 2021.

[35] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "CNN-generated images are surprisingly easy to spot... for now," arXiv [cs.CV], 2019.

[36] Y. Luo, Y. Zhang, J. Yan, and W. Liu, "Generalizing face forgery detection with high-frequency features," arXiv [cs.CV], 2021.

[37] I. Masi, A. Killekar, R. M. Mascarenhas, S. P. Gurudatt, and W. AbdAlmageed, "Two-branch recurrent network for isolating deepfakes in videos," in Computer Vision – ECCV 2020, Cham: Springer International Publishing, 2020, pp. 667–684.

[38] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "BlazeFace: Sub-millisecond neural face detection on mobile GPUs," arXiv [cs.CV], 2019.

[39] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," Information (Basel), vol. 11, no. 2, p. 125, 2020.

[40] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," arXiv [cs.LG], 2019.

[41] D. Kingma and J. Ba, "Adam: a method for stochastic optimization. arxiv: 14126980," 2014.

[42] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[43] P. Korshunov and S. Marcel, "Deepfake detection: humans vs. machines," arXiv [cs.CV], 2020.

[44] Y. Nirkin, L. Wolf, Y. Keller, and T. Hassner, "DeepFake detection based on the discrepancy between the face and its context," arXiv [cs.CV], 2020.

[45] H. Qi et al., "DeepRhythm: Exposing DeepFakes with Attentional Visual Heartbeat Rhythms," in Proceedings of the 28th ACM International Conference on Multimedia, 2020.

[46] A. Kumar, A. Bhavsar, and R. Verma, "Detecting deepfakes with metric learning," in 2020 8th International Workshop on Biometrics and Forensics (IWBF), 2020.

[47] H. Zhao, W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu, "Multi-attentional Deepfake Detection," arXiv [cs.CV], 2021.