Improving Generalization in Facial Manipulation Detection Using Image Noise Residuals and Temporal Features

Mehdi Atamna, Iuliia Tkachenko, Serge Miguet

IMPROVING GENERALIZATION IN FACIAL MANIPULATION DETECTION USING IMAGE NOISE RESIDUALS AND TEMPORAL FEATURES

Mehdi Atamna *Iuliia Tkachenko* *Serge Miguet*

Univ Lyon, Univ Lyon 2, CNRS, INSA Lyon, UCBL, Centrale Lyon,
LIRIS, UMR5205, F-69676 Bron, France

ABSTRACT

The high visual quality of modern deepfakes raises significant concerns about the trustworthiness of digital media and makes facial tampering detection more challenging. Although current deep learning-based deepfake detectors achieve excellent results when tested on deepfake images or image sequences generated using known methods, generalization—where a trained model is tasked with detecting deepfakes created with previously unseen manipulation techniques—is still a major challenge. In this paper, we investigate the impact of training spatial and spatio-temporal deep learning network architectures in the image noise residual domain using spatial rich model (SRM) filters on generalization performance. To this end, we conduct a series of tests on the manipulation methods of the FaceForensics++, DeeperForensics-1.0 and Celeb-DF datasets, demonstrating the value of image noise residuals and temporal feature exploitation in tackling the generalization task.

Index Terms— Deepfake detection, video manipulation detection, image forensics, steganalysis features

1. INTRODUCTION

The rapid progress in face swapping and facial reenactment technology has become a serious problem that facilitates the spread of manipulated media and misinformation, especially through the widespread availability of such tools and the wide reach of social media. In recent years, several methods have been proposed to effectively detect whether images or videos of people have been tampered with, and [1, 2] provide an overview of recent trends in the field of deepfake detection. In particular, deep learning-based detection methods have emerged as a popular tool, achieving excellent performance on state-of-the-art facial deepfake datasets [3, 4, 5, 6, 7, 8]. Convolutional neural network (CNN) architectures are especially popular, typically using a cropped image of the face of a subject in a video frame as input in order to classify it as real or fake [9, 4]. More recent work includes Transformer-type architectures repurposed for deepfake detection [10, 11]. Since current facial deepfakes are typically generated on a frame-by-frame basis and can thus exhibit temporal inconsistencies in a deepfake video, some deep architectures additionally leverage the temporal dimension by classifying *sequences* of images, for example by combining a CNN backbone with a recurrent architecture [3, 5] or by using 3D CNNs [8].

Aside from generic deep learning approaches, steganalysis and image forensics tools—which are typically used to detect forged images obtained through such methods as splicing—have also been applied to facial deepfake detection [4, 5]. The most effective of these methods [1], such as spatial rich model (SRM) [12] filters, leverage image noise residuals, the intuition being that synthetically manipulating a region in the image (i.e., the face) leads to local inconsistencies in the noise features of the modified image, even when the tampered image has been post-processed to better mask manipulation traces.

Although steganalysis models have been applied to the deepfake detection problem [4], notably as part of deep learning-based architectures [5, 7], their specific impact on generalization—the most challenging task as it exposes trained models to novel manipulation techniques—has not been extensively benchmarked. In this paper, we focus on training and evaluating the generalization performance of two types of architectures: XceptionNet [13], a popular CNN architecture for deepfake image classification which, despite its age, is still widely used for benchmarking purposes, and a spatio-temporal architecture combining XceptionNet and a long short-term memory (LSTM) network for image sequence classification in a cross-manipulation scenario in both the RGB image domain and the SRM noise residual domain. The problem is cast as a binary classification problem where the input data (images or image sequences depending on the architecture) is classified into real or fake. Cross-manipulation means that we train on real data as well as a group of manipulation techniques but withhold one manipulation type on which we evaluate generalization performance. We also study the impact of the length of the image sequence on generalization performance. We use FaceForensics++ [4], DeeperForensics-1.0 [14], and Celeb-DF [15], three state-of-the-art facial deepfake datasets in our tests. These datasets differ in scale and visual quality and can be broken down into three generations [16]: first (FaceForensics++), second

(Celeb-DF), and third (DeeperForensics-1.0).

The rest of the paper is organized as follows. In Section 2, we explain the baseline architectures used as well as the noise residual extraction method. In Section 3, we detail the data pre-processing and training procedures. In Section 4, we present the baseline architectures' test and generalization results and compare the color and noise residual domains in terms of performance. We also discuss the impact of image sequence length in this section. Finally, Section 5 concludes this article.

2. METHODS

2.1. Baseline architectures

In this work, we test our proposed approach in two different scenarios: (i) image classification, and (ii) image sequence classification.

For the former, we use an XceptionNet CNN that is pre-trained on ImageNet-1K [17], the fully connected layer being adapted to output scores for the two classes, *real* and *fake*. For sequence classification, we also use a pre-trained XceptionNet without the fully connected layer, where the last of the remaining layers' outputs are fed into a single-layer LSTM with 256 hidden units. The LSTM's output is used to classify the input sequence which, in this case, consists of five consecutive frames. It is important to note that grouping frames into sequences naturally results in fewer data points. Fig. 1 illustrates this spatio-temporal architecture.

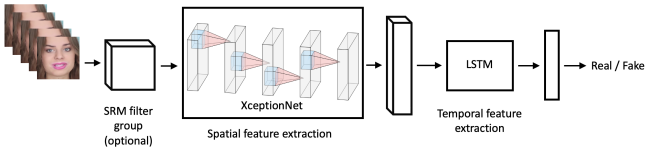

Fig. 1. The spatio-temporal architecture used.

2.2. Image noise residual computation

Modern facial deepfakes are typically post-processed in different ways with the goal of hiding visible tampering artifacts for a more photorealistic effect, which makes learning salient tampering features from RGB data exclusively more difficult. Noise, which is modeled here by the residual between a pixel's value and the estimate of said value produced by a linear combination of the neighboring pixels' values, can suppress high-level image content and highlight regions where an image has potentially been tampered with.

SRM filters are a representative method for the computation of image noise maps, also referred to as image noise residuals. Out of the thirty available SRM kernels, we select the same three as [18] to extract noise feature maps as, according to the authors, using the full thirty kernels does not improve performance significantly in their task, which is similar to deepfake detection. Fig. 2 shows these kernels.

Each kernel is expanded into three channels and applied to the input RGB image, which produces noise feature maps with the same dimensions as the RGB image once all three filters have been applied with appropriate padding of the image. This approach negates the need for architectural adjustments to the baseline networks, ensuring the comparability of results between the RGB and noise residual domains. Fig. 3 illustrates the noise feature maps obtained from selected images. We can see that most image content is suppressed and tampering traces in the eyes and the outline of the face are strongly highlighted.

$$\frac{1}{4}\times\begin{bmatrix}0&0&0&0&0\\0&-1&2&-1&0\\0&2&-4&2&0\\0&-1&2&-1&0\\0&0&0&0&0\end{bmatrix}\quad\frac{1}{12}\times\begin{bmatrix}-1&2&-2&2&-1\\2&-6&8&-6&2\\-2&8&-12&8&-2\\2&-6&8&-6&2\\-1&2&-2&2&-1\end{bmatrix}\quad\frac{1}{2}\times\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&1&-2&1&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$$

Fig. 2. The group of SRM filters used to compute noise features.

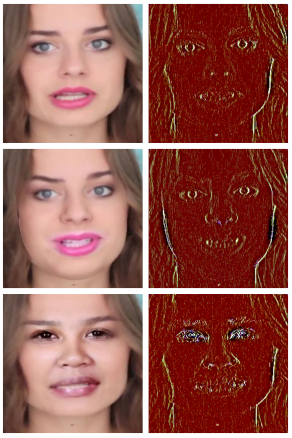

Fig. 3. The outputs of the selected SRM filter group (right) compared to the original FaceForensics++ images (left). Colors in the right column have been edited for better visibility. From top to bottom: real image, *Face2Face*, *FaceSwap* (see Section 3.1 for more details).

3. EXPERIMENTAL SETUP

3.1. Datasets

We use three state-of-the-art facial deepfake video datasets:

(i) FaceForensics++ (FF++): Contains 1,000 real videos and 5,000 fake videos made by editing the real videos using five different manipulation techniques. These manipulation techniques perform face swapping (*DeepFakes* (DF), *FaceShifter* (FSh), *FaceSwap* (FS)) and facial reenactment

(*Face2Face* (F2F), *NeuralTextures* (NT)). We use the H.264, lightly compressed (HQ) version of the dataset.

(ii) DeeperForensics-1.0 (DF-1.0): We use the standard (*std*) set which contains 1,000 fake videos obtained from the same non-manipulated videos as FaceForensics++ using the same HQ compression scheme.

(iii) Celeb-DF: Contains 5,639 fake and 590 real videos compressed to the MPEG4.0 format.

Train	Test	Xception-Net (RGB)	Xception-Net (SRM)	Xception-Net + LSTM (RGB)	Xception-Net + LSTM (SRM)
DF, F2F, FSh, FS, NT, DF-1.0		84.12	**93.43**	**97.73**	96.51
F2F, FSh, FS, NT, DF-1.0	DF	73.87	**83.16**	**91.94**	89.94
DF, FSh, FS, NT, DF-1.0	F2F	**63.68**	62.03	67.92	**73.44**
DF, F2F, FS, NT, DF-1.0	FSh	58.06	**65.19**	61.25	**69.93**
DF, F2F, FSh, NT, DF-1.0	FS	**47.27**	45.32	**47.79**	46.06
DF, F2F, FSh, FS, DF-1.0	NT	58.88	**62.26**	67.56	**70.29**
DF, F2F FSh, FS, NT	DF-1.0	55.05	**58.97**	**65.75**	62.92
DF, F2F, FSh, FS, NT	Celeb-DF	58.94	**60.97**	**61.40**	56.12
Celeb-DF	DF	**62.99**	61.13	**61.40**	58.42
	F2F	52.83	**54.25**	51.61	**54.89**
	FSh	**56.28**	49.96	**51.28**	45.46
	FS	50.33	**55.36**	49.78	**55.33**
	NT	**53.48**	52.56	**52.22**	50.39
	DF-1.0	52.63	**57.73**	**54.56**	51.40

Table 1. Classification results of both architectures. The first line corresponds to training and testing on the same manipulation techniques as well as real data, while the rest of the table considers the generalization scenario.

3.2. Data pre-processing

For each video, the first 120 frames are extracted, and the MTCNN [19] face detector is used to crop the subject's face in each frame. We manually verify the correct detection of the subject's face. The resulting crop is enlarged by 20% in both height and width to guarantee the presence of both manipulated and authentic regions then resized to 224×224. For image sequence classification, each sequence of five successive frames is taken as a data point, in a similar fashion to [5]. 75% of the data is used for training, 10% for validation, and 15% for testing. Data is split to make sure that content from any one video does not appear in more than one set—which could artificially inflate the results—and, in the case of FF++ and DF-1.0, the same IDs are used to split the real as well as the fake videos from all manipulation techniques into the various sets.

3.3. Training procedure

We use binary cross-entropy with appropriate class weights to account for the imbalance between real and fake data. We also use the Adam optimizer with a learning rate decay of 20% every 2,000 training steps. Classification accuracy is used as a performance metric, the same as in previous benchmarks [4]. In the image sequence classification task, accuracy is computed over all five-frame sequences, meaning that we do not compute scores for full videos.

For image classification, we train XceptionNet for five epochs with a batch size of 32 and an initial learning rate of 0.01. For image sequence classification, the spatio-temporal architecture is trained for eight epochs with a batch size of 16 and an initial learning rate of 0.001. Both architectures are pre-trained for three epochs where the XceptionNet weights from ImageNet-1K are frozen, allowing the fully connected and LSTM layers to learn while preserving the learned ImageNet features.

We train models on RGB images directly, then repeat the process using the same procedure and hyperparameters on noise residuals extracted with SRM filters.

4. RESULTS

4.1. Known manipulation techniques

Table 1 shows the classification results for the spatial and spatio-temporal architectures in both the RGB and noise residual domains (SRM). Since the fake videos of DF-1.0 (*std*) represent the same scenes as FF++, we group both datasets together in some tests. In the first line, the training and test sets contain real data as well as the listed manipulation types. This corresponds to the common scenario used in deepfake detection where models are evaluated on a test set containing real as well as manipulated data generated using methods seen during training. SRM significantly outperforms RGB training in image classification, but slightly falls behind in sequence classification. Note that the addition of temporal information is beneficial, as the spatio-temporal architecture outperforms XceptionNet overall.

This result shows that deep learning-based detection methods perform very well when tasked with classifying new data made using known manipulation techniques, as they are able to detect both real and manipulated data with relatively high accuracy.

4.2. Generalization performance

Generalization performance of the baseline architectures in a cross-manipulation scenario is shown starting from the second line of Table 1. Here, we evaluate the trained models on an unknown manipulation technique. For each test, we construct a perfectly balanced set of 150 videos of the unknown manipulation and 150 unseen real videos drawn from the same dataset.

We can see that training XceptionNet in the noise residual domain generally outperforms the RGB domain, in particular when numerous manipulation techniques are included in the training set. The results for the spatio-temporal architecture, however, are mixed. This can be explained by the fact that pre-processing the image with SRM filters removes various temporal manipulation artifacts such as abrupt illumination changes and color inconsistency between successive frames, which are very valuable in temporal analysis. High-frequency information is, as such, insufficient by itself in order to exploit all manipulation traces and allow the spatio-temporal architecture to learn rich features for tampering detection. This suggests that high-frequency feature extraction should ideally be viewed as a useful block to integrate into the design of a deep learning architecture which also exploits color features, rather than a standalone tool.

Also of note is the *FaceSwap* (FS) manipulation technique, which proves to be particularly challenging to detect. FS replaces a large area of the face and uses advanced post-processing steps [8], making it fundamentally different from other techniques, which may explain the poor generalization performance.

Sequence length	5	10	20	40	60
DF	61.40	69.92	62.28	65.44	**77.00**
F2F	51.61	66.08	53.06	55.00	**66.67**
FSh	51.28	50.28	51.44	**51.78**	48.00
FS	49.78	**55.64**	49.89	48.89	44.67
DF-1.0	54.56	**57.50**	53.72	53.33	47.17
NT	52.22	55.83	52.17	53.11	**62.50**
NT (train: FF++ & DF-1.0)	67.56	67.08	**70.50**	63.67	61.83

Table 2. Generalization results when varying the input sequence length (i.e., # successive images per sequence). Here, the spatio-temporal architecture is trained in the RGB domain on Celeb-DF, except in the last row.

4.3. Impact of image sequence length on generalization performance

In this section, we study the impact of image sequence length on generalization performance. We conduct a generalization test where the spatio-temporal architecture is trained on Celeb-DF multiple times, each time using a different sequence length, and evaluate on each of the remaining techniques. We train in the RGB domain only (i.e., without SRM) in this section. The results are shown in Table 2.

We can see that a sequence length of 60 successive frame can substantially improve generalization performance depending on the target manipulation. It is, however, important to note that longer sequences result in fewer data points for training and generalization testing, which may help explain why very long sequences are not always better. Results are also sensitive to training data, as evidenced by comparing NT results when using different training sets.

5. CONCLUSION

In this paper, we tackle the challenging deepfake generalization problem. We investigate the impact of training deep learning models in the image noise residual domain without modifying the neural network's architecture. We evaluate two different architectures: a CNN for image classification and a spatio-temporal architecture combining a CNN backbone with an LSTM for image sequence classification in a cross-manipulation scenario.

Through a series of generalization tests, we validate the viability of this approach and also show its limits across three state-of-the-art facial deepfake datasets: FaceForeniscs++, DeeperForensics-1.0, and Celeb-DF. Our experiments show that training on image noise residuals improves generalization performance on a majority of unseen manipulation types in the datasets we use in image classification. However, high-frequency information alone is insufficient for image sequence classification as other important clues, found in low-frequency image content, are suppressed.

We also investigate the impact of input sequence length on generalization performance, finding that longer input sequences can greatly improve generalization performance while being limited by sensitivity to the composition of the training set.

Future work will focus on the impact of video compression levels and schemes on generalization performance, and the integration of high-frequency processing pipelines into purpose-built deepfake detection architectures.

6. REFERENCES

[1] Luisa Verdoliva, "Media Forensics and DeepFakes: An Overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.

[2] Kratika Bhagtani, Amit Kumar Singh Yadav, Emily R. Bartusiak, Ziyue Xiang, Ruiting Shao, Sriram Baireddy, and Edward J. Delp, "An Overview of Recent Work in Multimedia Forensics," in *2022 IEEE 5th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2022, pp. 324–329.

[3] David Güera and Edward J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1–6.

[4] Andreas Rössler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1–11.

[5] Xi Wu, Zhen Xie, YuTao Gao, and Yu Xiao, "SSTNet: Detecting Manipulated Faces Through Spatial, Steganalysis and Temporal Features," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2952–2956.

[6] Davide Cozzolino, Andreas Rössler, Justus Thies, Matthias Nießner, and Luisa Verdoliva, "ID-Reveal: Identity-aware DeepFake Video Detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15088–15097.

[7] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu, "Generalizing Face Forgery Detection with High-frequency Features," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA, jun 2021, pp. 16312–16321, IEEE Computer Society.

[8] Ritaban Roy, Indu Joshi, Abhijit Das, and Antitza Dantcheva, "3D CNN Architectures and Attention Mechanisms for Deepfake Detection," in *Handbook of Digital Face Manipulation and Detection*, Springer International Publishing, Ed. 2022.

[9] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen, "MesoNet: a Compact Facial Video Forgery Detection Network," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2018, pp. 1–7.

[10] Davide Alessandro Coccomini, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi, "Combining EfficientNet and Vision Transformers for Video Deepfake Detection," in *Image Analysis and Processing – ICIAP 2022*. pp. 219–229, Springer International Publishing, 2022.

[11] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo, "Protecting Celebrities from DeepFake with Identity Consistency Transformer," *arXiv preprint arXiv:2203.01318*, 2022.

[12] Jessica Fridrich and Jan Kodovsky, "Rich Models for Steganalysis of Digital Images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[13] Francois Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[14] Liming Jiang, Ren Li, Wayne Wu, Chen Qian, and Chen Change Loy, "DeeperForensics-1.0: A Large-Scale Dataset for Real-World Face Forgery Detection," in *CVPR*, 2020.

[15] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for Deepfake Forensics," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3204–3213.

[16] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton-Ferrer, "The DeepFake Detection Challenge Dataset," *ArXiv*, vol. abs/2006.07397, 2020.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[18] Peng Zhou, Xintong Han, Vlad I. Morariu, and Larry S. Davis, "Learning Rich Features for Image Manipulation Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1053–1061.

[19] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.