DeepFake Cracker: a novel tool for deepfake video detection

Irene Amerini, Luca Maiano
Sapienza University of Rome, Italy

Roberto Caldelli
Universitas Mercatorum and CNIT, Italy

Leonardo Galteri, Alberto del Bimbo
University of Florence, Italy

***Abstract*—Recently, visual media technology has achieved impressive results; new tools for processing and, above all, generating multi-media contents have been developed. In particular, modern AI-based technologies have provided easy-to-use instruments to create extremely realistic manipulated videos. Such synthetic videos, named Deep Fakes, may constitute a serious threat to attack the reputation of public people or to address the public opinion on a certain event. According to this, being able to individuate such fake information becomes crucial. In this demo, the DeepFake Cracker is presented: a novel forensic tool able to discern between fake and original video sequences. Unlike other state-of-the-art methods which resort to single video frames to perform detection, we propose the adoption of optical flow fields to exploit possible inter-frame dissimilarities. Such a clue is then used as feature to be learned by CNN classifiers.**

I. Introduction

Deep learning techniques are escalating technology sophistication regarding creation and processing of multimedia contents. A new phenomenon, known as *Deep Fakes* (DF), has recently emerged: it permits to quite simply create realistic videos. In particular, people faces, or sometimes only lips and eyes movements, are substituted or modified in order to likely simulate the presence of another subject in a certain context or to make someone speak coherently with a different and, probably compromising, speech. The effects can be straightforwardly imagined when this fake information is deliberately used to harm a person such a public figure or a politician, or even an organization like a political party. The impact of Deep Fakes can also be amplified by the action of social networks that deliver information quickly and worldwide. According to this, machine learning community has dedicated a particular and twofold attention to this phenomenon. From one side, an effort has been spent to develop new kinds of effective synthesized video generation techniques such as Face2Face [1], Deep Video Portraits [2], StarGAN [3] and Deep Fake[1] among others. From another side, various studies have lastly focused on the problem to detect deepfake-like videos; most of them by analyzing possible inconsistencies within RGB frames of the video [4], [5], [6]. Usually, well established and pre-trained CNN techniques are directly applied to learn distinctive features from each single frame of the sequence. In [7], a recurrent convolutional strategy is used for face manipulation detection where a group of frames is evaluated as an ensemble. Other approaches consider physical characteristics, like eye blinking, [8] or biological signal [9]. In this demo, the *DeepFake Cracker* tool is presented exploiting a new technique able to detect deepfake-like videos from original ones. In particular, unlike state-of-the-art methods which usually act in a frame-based fashion, we employ a sequence-based approach dedicated to investigate possible dissimilarities in the temporal structure of a video. Specifically, optical flow fields have been extracted to exploit inter-frame correlations to be analyzed by a CNN classifier.

II. The reference method

The proposed approach [10] is based on *optical flow fields* and it exploits their capacity to distinguish a deepfake from an original video. Optical flow [11], [12] is a vector field which is computed on two consecutive frame $f(t)$ and $f(t+1)$ to extract apparent motion between the observer and the scene itself. In particular, the hypothesis behind such a method is that the optical flow is able to grab motion discrepancies across synthetically created frames with respect to those naturally generated by a video camera. It should be more appreciable in the optical flow matrices, the introduction of fake and unusual movements of the lips, eyes and in general of the whole face. So, for this reason, for each frame of a video sequence $f(t)$, at a certain time t, a forward flow $OF(f(t), f(t+1))$ is computed and passed, as input feature to be learnt, to a pre-trained CNN. In our experiments, we have tested *VGG16* [13], *ResNet50* [14] and *XceptionNet* [15] as backbones.

III. The demonstration tool

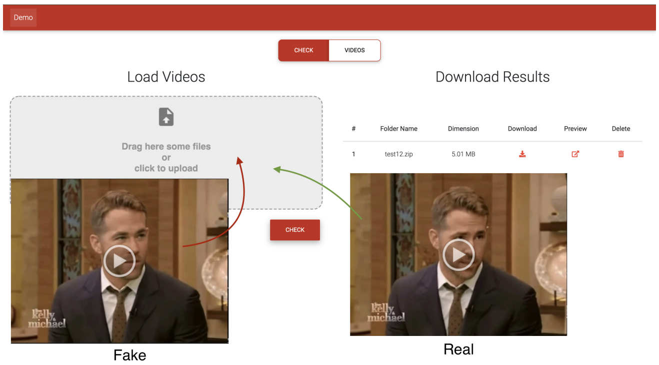
The DeepFake Cracker tool presented here, relies on our approach [10], based on optical flow, and drafted in the previous section. Such a tool is built on top of a trained model and will allow the users to put their hands on an actual detection phase of deepfake videos. In particular, in this case the convolutional neural network that has been selected is *ResNet50* which has been fine-tuned on the *Forensic++* dataset [5]. The demo is structured onto three main phases: the upload phase, the processing phase and the verification phase; all of them are described in the following subsections.

A. The upload phase

DeepFake Cracker presents an interface (see Figure 1) that basically permits, by means of a double button, to simply

[1] Deepfakes: github.https://github.com/deepfakes/faceswap.

switch between the video upload phase and the successive deepfake detection. Videos can be uploaded either by drag&drop or by browsing and then selecting a file. It is also possible to choose to upload multiple videos; this is particularly interesting when two videos that appear to be similar are to be compared and could constitute, for instance, the original source video and the deepfake one.

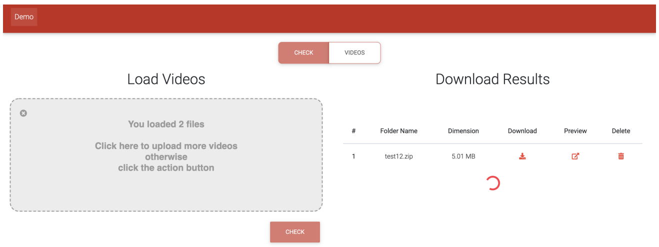


#	Folder Name	Dimension	Download	Preview	Delete
1	test12.zip	5.01 MB	⤓	↗	🗑

Fig. 1. The DeepFake Cracker interface: the upload phase

B. The processing phase

By switching to the processing phase (see Figure 2) by clicking on the button *check* on the top of the interface, it is possible to run the detection operation which starts with the optic flow extraction from the videos to be checked and proceeds by passing these features to the trained model which performs prediction. Detection results (frame-by-frame) are saved and can be downloaded for a more in-depth analysis. The optical flow computation is inevitably quite cumbersome with respect to the rest, so to grant execution times, compatible with the demo, such data can be calculated once for all and then charged according to the selected video to be analyzed.



#	Folder Name	Dimension	Download	Preview	Delete
1	test12.zip	5.01 MB	⤓	↗	🗑

Fig. 2. The DeepFake Cracker interface: the processing phase

C. The verification phase

The verification phase, reported in Figure 3, is the final step of the procedure. A pop-up window appears when the whole computation is completed; the frames that have been checked are now available and can be analyzed. The area around the face is highlighted by means of a squared bounding-box whose colour (red for fake and green for original) permits to visually understand what the system has predicted. In addition to this, a numerical predicted value (between 0 and 1) is also pictured. If the predicted value is close to zero the frame is labeled as fake, on the contrary is labeled as real. Finally the interface (see Figure 4) is able to provide the visualization of the results playing simultaneously the two videos (e.g. the deepfake one and the original one when it is possibly available) to better appreciate the visual differences between the two.

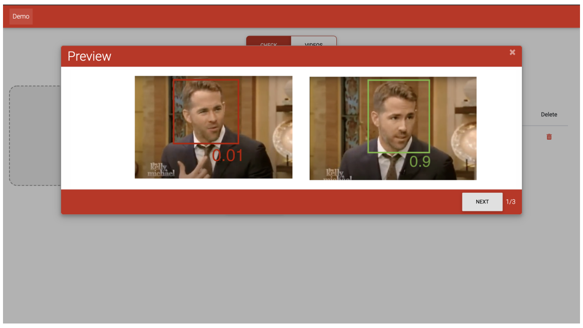


Fig. 3. The DeepFake Cracker interface: the verification phase

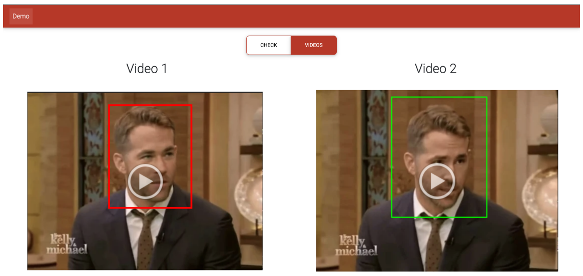


Fig. 4. The DeepFake Cracker interface: playing the result

IV. Implementation Details

The overall detector consists of three main parts: the Flask[2] component, the front-end (website) component and a Docker[3]. The flask folder contains the back-end code, written in Flask. Since an official Docker image of Flask does not exist, we created a specific one. So, inside the flask folder, there is a Dockerfile that is responsible for setting up our Flask image. The website folder contains the front-end code. Unlike Flask, for this purpose there is a pre-tested Docker image available on Docker Hub called httpd, so we do not need to set up our own custom Dockerfile. The docker file is in charge of merging the two above-described parts together, making them runnable from one single file. Inside this file, we declare our two services, namely the flask-service and the website. In other words, this file is responsible for building, on the one hand, the Flask image and, on the other hand, the httpd image.

[2] https://flask.palletsprojects.com/ [3] https://www.docker.com/

References

[1] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, "Demo of Face2Face: Real-time face capture and reenactment of RGB videos," in *ACM SIGGRAPH 2016 Emerging Technologies*, ser. SIGGRAPH '16. New York, NY, USA: ACM, 2016, pp. 5:1–5:2. [Online]. Available: http://doi.acm.org/10.1145/2929464.2929475

[2] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Niessner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt, "Deep video portraits," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 163:1–163:14, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3197517.3201283

[3] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," *CoRR*, vol. abs/1711.09020, 2017. [Online]. Available: http://arxiv.org/abs/1711.09020

[4] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics: A large-scale video dataset for forgery detection in human faces," *CoRR*, vol. abs/1803.09179, 2018. [Online]. Available: http://arxiv.org/abs/1803.09179

[5] ——, "Faceforensics++: Learning to detect manipulated facial images," *CoRR*, vol. abs/1901.08971, 2019. [Online]. Available: http://arxiv.org/abs/1901.08971

[6] 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)*, 12 2018, pp. 1–7.

[7] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," 05 2019.

[8] Y. Li, M. Chang, and S. Lyu, "In ictu oculi: Exposing AI generated fake face videos by detecting eye blinking," *CoRR*, vol. abs/1806.02877, 2018. [Online]. Available: http://arxiv.org/abs/1806.02877

[9] U. A. Ciftci, I. Demir, and L. Yin, "Fakecatcher: Detection of synthetic portrait videos using biological signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1–1, 2020. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2020.3009287

[10] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake video detection through optical flow based cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[11] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Comput. Surv.*, vol. 27, no. 3, pp. 433–466, Sep. 1995. [Online]. Available: http://doi.acm.org/10.1145/212094.212141

[12] L. Alparone, M. Barni, F. Bartolini, and R. Caldelli, "Regularization of optic flow estimates by means of weighted vector median filtering," *IEEE Transactions on Image Processing*, vol. 8, no. 10, pp. 1462–1467, Oct 1999.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[15] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," *CoRR*, vol. abs/1610.02357, 2016. [Online]. Available: http://arxiv.org/abs/1610.02357