

# Dynamic texture analysis for detecting fake faces in video sequences

Mattia Bonomi<sup>†</sup>, Cecilia Pasquini and Giulia Boato

**Abstract**—The creation of manipulated multimedia content involving human characters has reached in the last years unprecedented realism, calling for automated techniques to expose synthetically generated faces in images and videos.

This work explores the analysis of spatio-temporal texture dynamics of the video signal, with the goal of characterizing and distinguishing real and fake sequences. We propose to build a binary decision on the joint analysis of multiple temporal segments and, in contrast to previous approaches, to exploit the textural dynamics of both the spatial and temporal dimensions. This is achieved through the use of Local Derivative Patterns on Three Orthogonal Planes (LDP-TOP), a compact feature representation known to be an important asset for the detection of face spoofing attacks.

Experimental analyses on state-of-the-art datasets of manipulated videos show the discriminative power of such descriptors in separating real and fake sequences, and also identifying the creation method used. Linear Support Vector Machines (SVMs) are used which, despite the lower complexity, yield comparable performance to previously proposed deep models for fake content detection.

## I. INTRODUCTION

Being able to ensure and verify the integrity of digital multimedia content is recognized as an essential challenge in our society. In the last decade, the field of multimedia forensics has worked towards developing increasingly effective technological safeguards to address these issues, with the goal of inferring information on the acquisition settings and digital history of the images and videos under investigation.

In parallel, computer graphics and machine vision have achieved impressive advances in the very last years in the creation of highly realistic synthetic audio-video content. Convincing digital representations of human characters appearing almost indistinguishable from real people can now be obtained automatically through increasingly accessible tools. These technologies are progressing at a tremendous pace, and can be coupled with advances in the field of text-to-speech synthesis. While offering exciting opportunities for entertainment and content creation purposes, it is clear that such technologies can have significant security implications in different application scenarios. As a matter of fact, digital versions of human faces are constantly streamed through video chats, video conferencing services, media channels, and even

used for authentication purposes in replacement of traditional schemes based on fingerprints or passwords.

Thus, the need for forensic techniques able to deal with this new powerful manipulations has become of primary importance, leading to huge efforts and initiatives in developing robust forensic detection methodologies and benchmarking them on common datasets [40], [17]. While the identification of computer-generated faces has been widely addressed in the last decade, the data produced by advanced and AI-based creation tools have raised renovated attention due to the higher level of hyper-realism [47], as well as new and more complicated technical challenges. In response, a number of new detection approaches have been proposed, with special focus on still images depicting synthetic faces. However, the problem of detecting fake characters in video sequences has been faced only very recently, since the quality of AI-generated videos depicting faces achieved only in the last couple of years a good level of perceptual quality and realism. Currently, video forensics approaches developed for this problem mostly apply detection techniques designed for still images to single frames of the video sequence, often relying on deep representations of the pixel domain. However, in doing so they do not exploit the temporal information provided by video sequences, which might contain useful statistical characterizations and contribute to the detection capabilities of an automatic detector.

The analysis of discriminative cues over time is tackled by a few previous works. One direction is to detect behavioural anomalies of the face dynamics, like the absence of physiological signals [11], inconsistent expression patterns [4], irregular eye-blinking [27]. While in principle these methods are robust to geometric degradations and easily interpretable, their effectiveness is highly dependent on the scene content, as it is based on few semantic cues that might not be available in all video sequences. Also, deep learning machinery (like recurrent neural networks [41], [22], [5]) has very recently been used to model short frame sequences, showing promising results at the price of low interpretability, a typical issue of deep learning based approaches. Moreover, deep learning based techniques present the classical drawback of requiring careful training on a large and diverse amount of data to achieve transferability of results and to avoid overfitting.

In this work, we aim at exploiting both texture and temporal information of the video sequence, by tackling an intermediate approach that relies on hybrid descriptors operating in both spatial and time domain. This yields relatively small feature representations that can be learned through simpler classifiers, such as linear SVMs. While such descriptors have been successfully used for video-based face spoofing detection [43],

<sup>†</sup> Corresponding author.

M. Bonomi, C. Pasquini and G. Boato are with the Department of Information Engineering and Computer Science, University of Trento, Trento 38123, Italy (e-mail: mattia.bonomi@unitn.it, giulia.boato@unitn.it). C. Pasquini was with the Department of Computer Science, Universität Innsbruck, Innsbruck 6020, Austria (e-mail: cecilia.pasquini@uibk.ac.at).

to the best of our knowledge their effectiveness has never been explored in the context of manipulated faces detection, although the two problems present significant analogies. Our approach employs so-called Local Derivative Patterns on Three Orthogonal Planes (LDP-TOP), a variant of local binary patterns that operates on three dimensions and proved to be particularly effective in face anti-spoofing. Moreover, we propose to perform the analysis of entire video sequences by combining the predictions computed on multiple temporal segments, which proves to bring a significant accuracy gain.

The remainder of the paper is structured as follows: Section II summarizes the state of research for the problem of manipulated video and image faces detection; in Section III, we illustrate how the feature descriptors are extracted from single videos, while Section IV describes the proposed classification framework. Experimental results are reported in Section V and conclusions are drawn in Section VI.

## II. DETECTION OF MANIPULATED FACES IN IMAGES AND VIDEOS

In order to position our work with respect to existing literature, this section briefly reviews the main classes of methods proposed for detecting manipulated and computer-generated (CG) faces in multimedia data.

### A. Methods based on statistical hand-crafted features

Several methods proposed to distinguish real from manipulated multimedia content by exploiting statistical features capturing intrinsic properties of the media object. Earlier works study specific traces that are present in real data due to operations at acquisition time [33], such as color filter array interpolation [21], or lens chromatic aberration [15]. Other approaches extract statistical features capturing the characteristics of the spatial texture [36] [25] and the coefficients distribution in transformed domains (e.g., wavelet) [29] [9], leading to supervised classification frameworks combining these cues [37].

More recently, detectors based on Fourier analysis coupled with conventional machine learning have been proposed also for modern AI-based manipulations [16]. Such methods are applied to images only, thus they do not deal with the temporal evolution of video signals. As detailed in Section III, our work fills this gap by proposing a spatio-temporal texture description.

### B. Methods based on deep neural networks

Deep neural networks are not only used for creation purposes but also as powerful tools for detecting fake content.

Several studies have been conducted on the use of deep networks to detect fake images generated by Generative Adversarial Networks (GANs) [31], [53], [30], and identify fingerprint specific GANs may leave [52].

A number of Convolutional Neural Networks (CNN) architecture have been proposed for the detection of manipulated faces videos, with the goal of characterizing artifacts arising when generating fake content. The authors in [39], [3] propose two shallow CNNs architectures exploiting mesoscopic features. In [40], it is shown that deeper general-purpose networks

like XceptionNet in the same supervised scenario generally outperform shallow ones, as well as where re-adapted feature-based methods originating from steganalysis [20] and general-purpose image forensics [7].

While these methods are applied individually on video frames, only few works operate along the temporal dimension. This is done in [41] and [22] through the use of recurrent neural networks. In [5], a CNN is used to estimate and analyze the optical flow field across frames.

Finally, several deep-learning techniques have been recently proposed for other security applications, including the analysis of surveillance videos [49], and the detection of suspect videos through usage of blockchain and smart contracts [23].

While the mentioned approach deliver good results in supervised scenarios, they typically tend to overfit the training set and suffer from performance decrease when dealing with unseen manipulations [26]. Additional strategies are then necessary to increase generalization capabilities, such as attention mechanisms [44] or segmentation modules [34]. We refer the reader to [46] [35] for thorough surveys of the literature on the topic.

### C. Methods based on semantic cues

As an alternative to hand-crafted or self-learned features, a number of methods aims at characterizing semantic features differentiating real and manipulated content. The work in [32] extracts typical artifacts appearing in GAN-generated images, such as non symmetrical colors and shape (in eyes and ears) or badly rendered details (e.g., blurry teeth areas).

Earlier studies on rendered faces exploited geometric properties of the face in the spatial [12] and temporal domain [13]. Further properties like inconsistencies in facial landmark locations [51], head pose [50], and eye-blinking [28] have also been exploited for exposing fakes.

By relying on video magnification techniques [48], the techniques developed in [11] and [8] estimate the pulse rate of the depicted subject from temporal skin color variations, and show that this physiological signal is typically flat when the subject face is manipulated or computer generated. Similar ideas are explored in [10] and [18], where deep networks are used for this purpose.

Moreover, recent approaches [4] study and characterize soft traits specific individuals have in reproducing facial expressions and head movements, which are hardly reproducible in manipulated content.

## III. EXTRACTION OF SPATIO-TEMPORAL TEXTURAL FEATURES

The methodology proposed in this work is composed of a preprocessing phase and a feature extraction phase. These two processes are described in the following subsections. Obtained feature representation will be learned in the classification phase described in the next section.

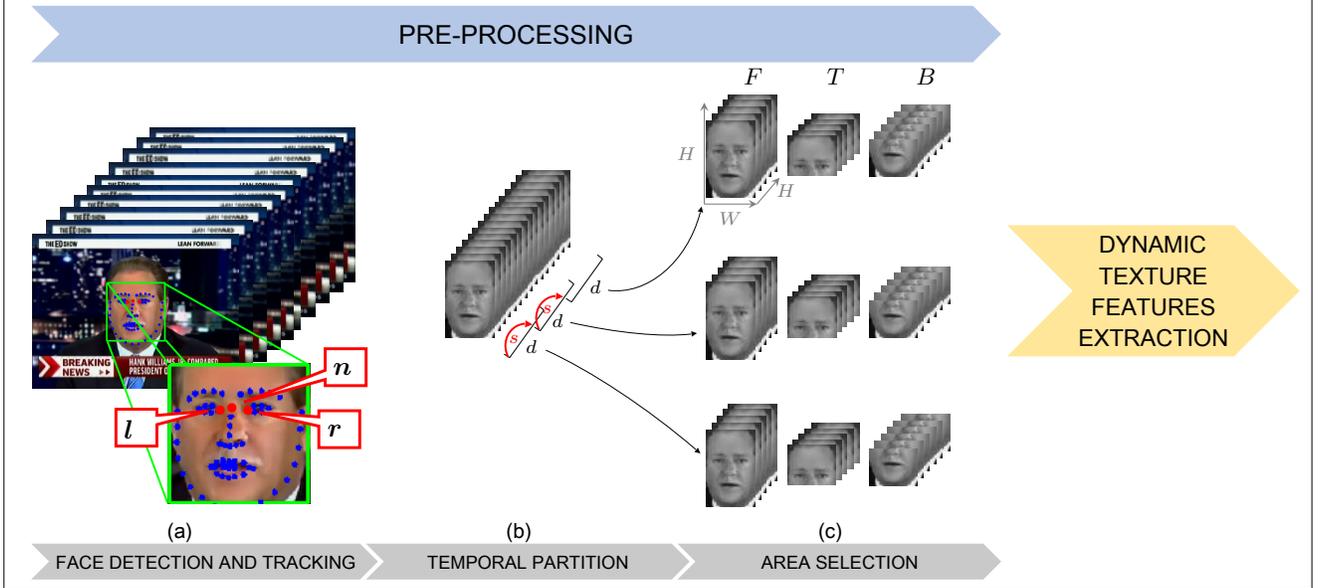


Fig. 1: Workflow of the proposed pre-processing pipeline.

### A. Pre-processing

First, video patches are extracted and partitioned in multiple temporal sequences<sup>1</sup>. The different steps involved in the pre-processing pipeline are depicted in Fig. 1 and explained below:

- (a) *Face detection and tracking*: after extracting the frames, the Python library `dlib` (v. 19.8.1) is used on the first video frame to obtain the ROI patch containing the face, as well as on every subsequent frame to detect the 68 facial landmarks. The three landmarks corresponding to the right eye lacrimal caruncle ( $r$ ), the left eye lacrimal caruncle ( $l$ ), and top nose ( $n$ ) are selected. A motion vector  $\Delta$  is then computed between each pair of consecutive frames by averaging the horizontal and vertical displacements of  $r$ ,  $l$  and  $n$ , and smoothed temporally

<sup>1</sup>Python 3.6.7 with the OpenCV2 4.1.0 libraries and MATLAB R2019a have been used for the implementation.

through a Savitzky-Golay filter on both dimensions [42]. The initial patch is then tracked over time by shifting it of  $\Delta$  frame by frame.

- (b) *Temporal partition*: after conversion to grayscale, overlapping temporal windows of  $d$  seconds with a stride of  $s$  seconds are isolated. This yields different temporal sequences of frames, whose numerosity depends on the duration of the video. A generic temporal sequence  $S$  resulting from this process is a 3D array of pixels of size  $H \times W \times K$ , where  $H$  and  $W$  depend on the output of the face detector on the first frame, and  $K$  depends on the frame rate of the video.
- (c) *Area selection*: at this stage, we allow to select a specific area of the face to be used for the feature analysis, in order to observe the relevance of different regions for the chosen feature representation. In our tests, we have considered three different cases, denoted in the following with upper-case letters (see Fig. 1): the top-half ( $T$ ), the bottom-half ( $B$ ), or the full face information ( $F$ ) is used.

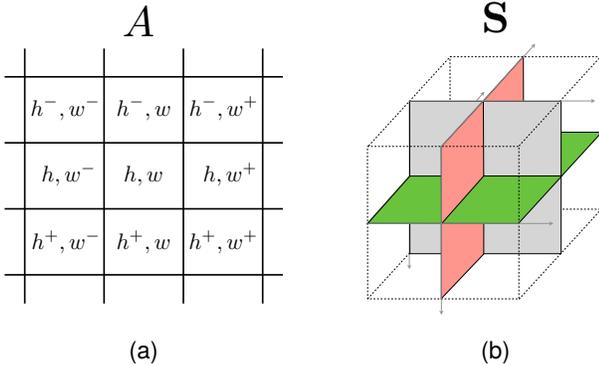


Fig. 2: Representations of the  $3 \times 3$  neighborhood and the three orthogonal planes used for the extraction of the LDP-TOP descriptors.

### B. Dynamic texture features

We aim at exploiting both spatial and temporal domains in the analysis of video sequences. To this purpose, we considered the Local Derivative Pattern features (LDP), already used for face recognition as a pattern descriptor (e.g. [24], [6]), in their extended version involving the temporal domain, the Local Derivative Pattern on Three Orthogonal Planes (LDP-TOP) [38].

The LDP, a generalization of the widely used Local Binary Pattern (LBP), is a point-wise operator applied to 2D arrays of pixels, that encodes diverse local spatial relationships. As suggested in [6], we consider the second-order directional LDPs with direction  $\alpha$ , indicated as  $LDP_{\alpha}^2$ , where  $\alpha \in \{0^{\circ}, 45^{\circ}, 90^{\circ}, 135^{\circ}\}$ . Given a 2D array of pixels  $A$ , the  $LDP_{\alpha}^2$  at the location  $(h, w)$  is an 8-bit vector defined as:

$$\begin{aligned} \text{LDP}_\alpha^2(h, w) = & [f(I'_\alpha(h, w), I'_\alpha(h^-, w^-)), f(I'_\alpha(h, w), I'_\alpha(h^-, w)), \\ & f(I'_\alpha(h, w), I'_\alpha(h^-, w^+)), f(I'_\alpha(h, w), I'_\alpha(h, w^+)), \\ & f(I'_\alpha(h, w), I'_\alpha(h^+, w^+)), f(I'_\alpha(h, w), I'_\alpha(h^+, w)), \\ & f(I'_\alpha(h, w), I'_\alpha(h^+, w^-)), f(I'_\alpha(h, w), I'_\alpha(h, w^-))] \end{aligned}$$

with  $h^+ := h+1, h^- := h-1$  and  $w^+ := w+1, w^- := w-1$ . A representation of the  $3 \times 3$  neighborhood is depicted in Figure 2(a). The operator  $I'_\alpha$  is the first-order derivative in the direction  $\alpha$ , and is defined pixel-wise as:

$$I'_\alpha(h, w) = \begin{cases} A(h, w) - A(h, w^+) & \text{if } \alpha = 0^\circ \\ A(h, w) - A(h^-, w^+) & \text{if } \alpha = 45^\circ \\ A(h, w) - A(h^-, w) & \text{if } \alpha = 90^\circ \\ A(h, w) - A(h^-, w^-) & \text{if } \alpha = 135^\circ \end{cases} \quad (1)$$

while

$$f(a, b) = \begin{cases} 0 & \text{if } x \cdot y > 0 \\ 1 & \text{if } x \cdot y \leq 0 \end{cases} \quad (2)$$

Essentially,  $\text{LDP}_\alpha^2(h, w)$  encodes whether first-order derivatives in the direction  $\alpha$  have consistent signs when computed at  $(h, w)$  and at proximal pixel locations. For a 2D array, the  $\text{LDP}_\alpha^2$  are extracted for every pixel and their  $2^8$ -bin histogram is computed; this is replicated for the four different directions, and the histograms are concatenated.

Similarly as it is done in [19] for LBPs, in [38] the authors propose to extend the computation of LDP histograms to 3D arrays. This is done by sequentially considering the three central 2D arrays along each dimension that intersect orthogonally (see Figure 2(b)) and again concatenating the obtained histograms, yielding the so-called LDP-TOP features.

In our case, we apply this procedure to the temporal sequences  $\mathcal{S}$  extracted as in Section III-A, and use the obtained histograms as features. Considering 4 derivative directions and three 2D arrays, the feature vector length is equal to  $2^8 \times 4 \times 3 = 3072$ .

In order to explore potential peculiarities in the way the temporal information is captured by LDPs, we add the opportunity to run the feature extraction on  $\mathcal{S}$  in three different temporal modes, which differ by the orientation of the temporal information. In particular, we define:

- *Direct mode* ( $\rightarrow$ ):  $\mathcal{S}$  is processed forward along the temporal direction;
- *Inverse mode* ( $\leftarrow$ ):  $\mathcal{S}$  is processed backward along the temporal direction starting from the last frame;
- *Bidirectional mode* ( $\leftrightarrow$ ):  $\mathcal{S}$  is processed in both directions and histograms are concatenated (thus yielding a feature vector with doubled size).

#### IV. CLASSIFICATION FRAMEWORK

We now describe the framework adopted in our study for training a classifier and taking a decision on single tested videos.

As depicted in Fig. 3, the training process involves a set of real and manipulated videos, that we indicate as  $\mathcal{TR}_r$  (labeled as 0) and  $\mathcal{TR}_m$  (labeled as 1), respectively. Every video in

these sets is fed into the pre-processing and the descriptor computation blocks, as described in Sections III-A and III-B. The feature vectors computed from each temporal sequences inherit the label of the video they belong and all of them are used as inputs for training the classifier  $C$ , a Support Vector Machines (SVM) with linear kernel<sup>2</sup>.

Afterwards, the videos to be tested belong to sets that we will indicate as  $\mathcal{TS}_r$  and  $\mathcal{TS}_m$ . The prediction on single videos is computed as depicted in Fig. 4. Pre-processing and descriptor computation are again performed and each resulting feature vector extracted is passed to the trained SVM model. This returns a pair  $p_n, s_n$  for each of the  $N$  temporal sequences extracted, where  $p_n$  is the predicted label and  $s_n$  is the output score of the SVM. In order to determine a final label  $\hat{p}$  for the input video, a majority voting criterion is employed:

$$\hat{p} = \text{maj}(\{p_1, \dots, p_N\}) \quad (3)$$

where  $\text{maj}(\cdot)$  outputs the value recurring most frequently in the input set. In case of equal number of conflicting predictions, the  $\text{maj}$  criterion conservatively favors the 0 class.

Finally, for each video we compute a final score  $\hat{s}$  through a “reduced mean” criterion:

$$\hat{s} = \text{mean}(\{s_n \text{ where } n \text{ is such that } p_n = \hat{p}\}), \quad (4)$$

i.e., only the score values corresponding to the sequences whose predictions correspond to the final prediction  $\hat{p}$  are averaged.

#### V. EXPERIMENTAL RESULTS

The next sections present the experimental tests conducted in order to validate the proposed method in practical scenarios.

As a benchmark dataset of real and fake videos, we considered the FaceForensics++ dataset described in [40], which consists of a large set of videos depicting human faces, which are then manipulated with different techniques. In particular, we have considered the 1000 original videos (OR) and their manipulated counterparts through the *Deepfake* (DF) [1], the *Face2Face* (F2F) [45] and the *FaceSwap* (FSW) [2] techniques. We operate on the version of the dataset subject to a light compression (H.264 with constant rate quantization parameter equal to 23). An example of these different manipulations is depicted in Fig. 5. The videos are recorded under different conditions (e.g., interviews, TV shows, etc.), they have different length and are captured by different cameras. This results into a huge variability in terms of both data content and video structure (i.e., frame rate, video length, original coding standards, etc).

The dataset comes with a standard split of videos for training, validation, and testing. In order to enable a fair comparison with other recently proposed approaches, we also considered the same training and testing set, yielding the sets  $\mathcal{TR}_D$  with  $|\mathcal{TR}_D| = 720$  and  $\mathcal{TS}_D$  with  $|\mathcal{TS}_D| = 140$ , where  $D \in \{\text{OR}, \text{DF}, \text{F2F}, \text{FSW}\}$ . Different subsets will be combined according to the experimental scenario considered.

<sup>2</sup>We used the MATLAB Statistics and Machine Learning Toolbox (v. R2019a) and selected a linear kernel function with predictor data standardization and Sequential Minimal Optimization (SMO).

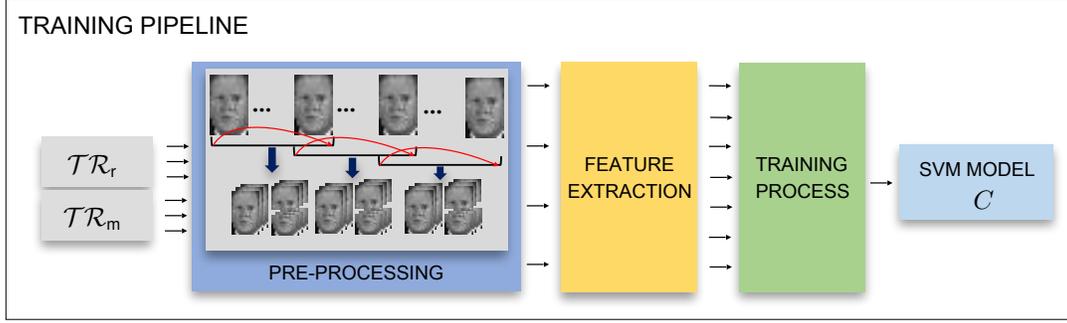


Fig. 3: Training pipeline: given as input the training set of real and fake videos, provides as output the corresponding SVM model.

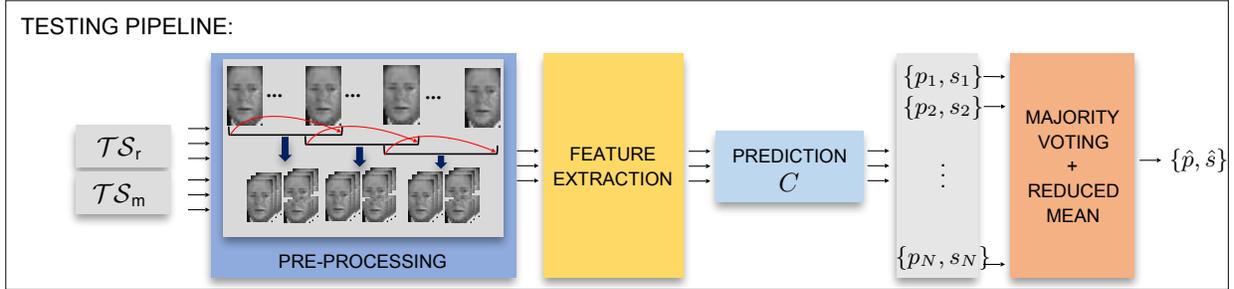


Fig. 4: Testing Pipeline: the pipeline  $C$  returns a binary label  $\hat{p}$  and the corresponding score  $\hat{s}$ .

We have tested the feature representation and classification framework proposed in Section III and IV in several experimental scenarios and by analyzing different factors, which are described in details in the next subsections. For the sake of readability, we first summarize here the structure of our experimental validations:

- **Single-technique scenario** (Section V-A). Original and fake videos are considered separately for different creation techniques; the impact of the temporal partition operation, the face area selection, and the temporal mode adopted are discussed.
- **Multiple-technique scenario** (Section V-B). Videos created with arbitrary manipulation techniques are mixed in the testing; the capabilities of detecting and identifying

the manipulation technique used in the testing phase is evaluated.

- **Strong video compression** (Section V-C). The proposed detector is tested when a heavier compression is applied to the videos, thus its robustness against video compression is analyzed.
- **Comparison with other descriptors** (Section V-D). Performance comparison is discussed both considering the proposed detector exploiting the alternative spatio-temporal feature representation given by the LBP-TOP and other SoA approaches.

#### A. Single-technique scenario

We tested the performance of our approach in separating original videos from videos that have been manipulated with a specific technique. The goal is to show the capabilities of each classifier when subjected to its corresponding test set. Thus:

$$\begin{aligned} TR_r &= TR_{OR} & TR_m &= TR_D \\ TS_r &= TS_{OR} & TS_m &= TS_D \end{aligned}$$

where  $D$  varies in the set  $\{DF, F2F, FSW\}$ . This yields an SVM classifier for every manipulation technique, that we denote as  $C_{DF}$ ,  $C_{F2F}$  and  $C_{FSW}$ .

Videos in these sets are fed into the training pipeline described in Fig. 3. In this phase, we report the results obtained by employing the three different facial areas ( $F$ ,  $T$ , and  $B$ ) specified in Section III-A and the three temporal modes ( $\rightarrow$ ,  $\leftarrow$ ,



Fig. 5: Frames extracted from (a) a sample OR video sequence and its (b) DF, (c) F2F and (d) FSW manipulated versions.

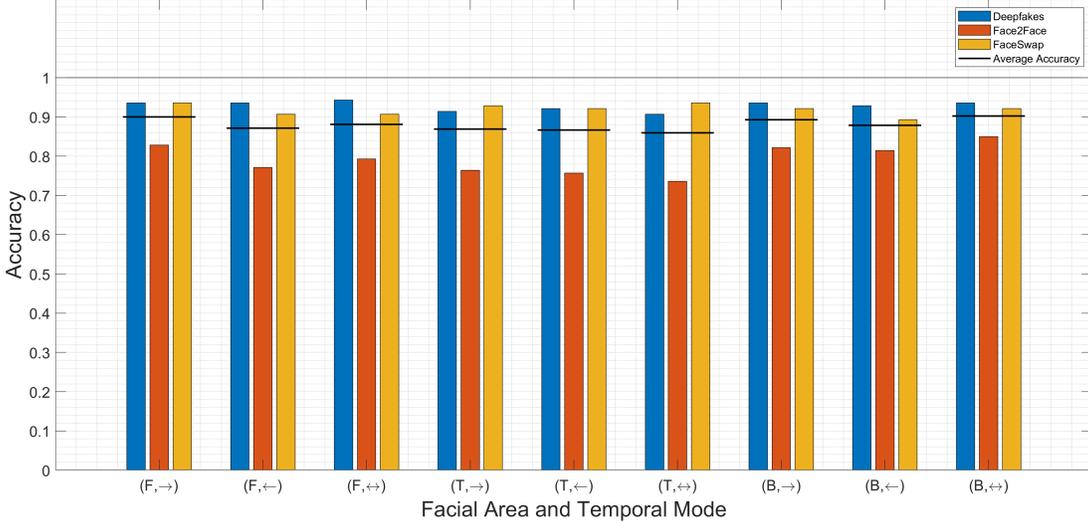


Fig. 6: Classification accuracy per manipulation technique.

TABLE I: Classification accuracy and AUC computed on the single-manipulation scenario. Different facial areas and temporal modes are considered.

Algorithm Version	Accuracy			Average Accuracy	AUC			Average AUC
	Deepfakes	Face2Face	FaceSwap	Cross-Dataset	Deepfakes	Face2Face	FaceSwap	Cross-Dataset
(F, →)	93,57%	82,86%	93,57%	90,00%	98,23%	88,08%	98,22%	<b>94,85%</b>
(F, ←)	93,57%	77,14%	90,71%	87,14%	98,78%	85,14%	98,00%	93,97%
(F, ↔)	94,29%	79,29%	90,71%	88,10%	98,65%	86,94%	97,45%	94,35%
(T, →)	91,43%	76,43%	92,86%	86,90%	95,39%	78,13%	98,18%	90,57%
(T, ←)	92,14%	75,71%	92,14%	86,67%	94,41%	78,39%	98,00%	90,27%
(T, ↔)	90,71%	73,57%	93,57%	85,95%	94,89%	80,78%	98,06%	91,24%
(B, →)	93,57%	82,14%	92,14%	89,29%	97,53%	86,64%	97,47%	93,88%
(B, ←)	92,86%	81,43%	89,29%	87,86%	97,57%	85,91%	97,55%	93,68%
(B, ↔)	93,57%	85,00%	92,14%	<b>90,24%</b>	97,55%	88,63%	97,47%	94,55%

and  $\leftrightarrow$ ) specified in Section III-B, yielding to nine classifiers per manipulation technique, to observe how they vary and interact.

Results are depicted as bar plots in Figure 6 in terms of accuracy, i.e., the fraction of videos in  $\mathcal{TS}_f \cup \mathcal{TS}_m$  that is assigned to the correct label. Full numerical results are reported in Table I, where the value of the Area Under the Curve (AUC) obtained by thresholding  $\hat{s}$  (i.e., the reduced-mean score) is also reported as performance indicator.

Tab. I suggests that  $C_{DF}$  and  $C_{FSW}$  almost always allow for an accuracy greater than 90%, while for  $C_{F2F}$  the accuracy does not exceed 85,0%. Interestingly, this correlates with the observations made in [40], where a user study reveals that F2F generally produces more challenging manipulations to be detected for humans.

Moreover, it can be noticed that both the  $F$  and the  $B$  facial areas versions provide a better accuracy with respect to  $T$ . This indicates that the artifacts captured by the proposed feature representation are generally concentrated in the bottom part of the face. However, this effect is not uniform across manipulation techniques (see FSW), suggesting that manipulation-specific patterns are likely introduced, as we will exploit in the next subsection.

Finally, we observe that the inverse temporal mode alone does not introduce significant advantages, while the bidirectional mode generally does. This is not so surprising, given

that the feature vector size is doubled, however the number of training samples remains the same.

In summary, the best results in terms of both performance indicators are achieved in the  $(F, \rightarrow)$  and the  $(B, \leftrightarrow)$  cases, respectively yielding 90,00% and 90,24% average accuracy. Therefore, for the sake of readability and space, we focus on the corresponding classifiers for the experimental analyses in the next subsections.

As a further analysis, we evaluate the benefits of applying the temporal partition through sliding windows in the preprocessing phase by comparing with the baseline case where videos are not subdivided in shorter video sequences (i.e., the  $w$  parameter in Fig. 1 is set equal to the video length in seconds) and only one LDP-TOP feature vector is extracted from each single video. This corresponds to the common approach of previously proposed detection methods (see [40]).

First, we observe in Table II how the number of input feature vectors changes for these two cases, noticing that the sliding window approach increases the number of training/testing feature vectors by 6 to 8 times. Then, we provide in Table III the accuracy loss when skipping the temporal partition step, defined as the difference in accuracy between of the “sliding” and “non-sliding” case (i.e., positive values indicate better performance of the “sliding” case). It can be noticed that the “sliding” approach always outperforms the “non-sliding” in

TABLE II: Comparison between the number of samples (batches) obtained in case of non-sliding and sliding window approaches.

	OR		DF		F2F		FSW	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Sliding	3029	588	3026	588	2966	640	2307	482
Non-sliding	360	70	360	70	360	70	360	70

TABLE III: Classification accuracy loss per manipulation technique when applying the "non-sliding" approach.

Algorithm Version	Accuracy Loss			Average Accuracy Loss
	DF	F2F	FSW	Cross-Dataset
$(F, \rightarrow)$	1,43%	10,72%	-2,14%	3,33%
$(F, \leftarrow)$	2,14%	5,71%	-2,15%	1,90%
$(F, \leftrightarrow)$	2,15%	6,43%	-4,29%	1,43%
$(T, \rightarrow)$	2,14%	10,00%	0,00%	4,04%
$(T, \leftarrow)$	2,14%	3,57%	0,00%	1,91%
$(T, \leftrightarrow)$	1,42%	-1,43%	2,14%	0,71%
$(B, \rightarrow)$	-0,72%	3,57%	0,00%	0,96%
$(B, \leftarrow)$	0,00%	5,72%	-2,85%	0,96%
$(B, \leftrightarrow)$	1,43%	7,14%	0,00%	2,86%

terms of average accuracy among all datasets, with significant improvements (up to 10%) for F2F. Just in some single cases, especially for FSW, this observation is reversed, showing again manipulation-specific peculiarities. The two selected classifiers (top and bottom one in Table III) however adhere to the general trend, showing an average accuracy increase of 3,33% and of 2,86%.

### B. Multiple-technique scenario

We now consider the case where manipulation techniques are mixed. In particular, we approach the more realistic case where

$$TS_r = TS_{OR} \quad TS_m = TS_{DF} \cup TS_{F2F} \cup TS_{FSW}$$

and the binary decision on each testing video needs to be taken blindly, i.e., without prior information on the manipulation technique used.

We have experienced that training a single binary classifier with  $TR_r = TR_{OR}$  and  $TR_m = TR_{DF} \cup TR_{F2F} \cup TR_{FSW}$  brings to poor results. This might be interpreted in view of the linearity of the classifier used, which seemingly does not allow to properly separate the two classes through an hyperplane in the feature space. Instead of enforcing that a single classifier can accurately separate the samples, we rather propose to combine the outcome of classifiers trained on single manipulation techniques. This also allows us to estimate the used manipulation technique in case of positive detection in a cascade fashion as represented in Figure 7.

More specifically, we propose to assign each test video a label  $\hat{p} \in \{0, 1\}$  by combining the outputs of the classifiers  $C_{DF}$ ,  $C_{F2F}$  and  $C_{FSW}$  trained as in Section V-A. This yields to three predicted labels  $\hat{p}_{DF}$ ,  $\hat{p}_{F2F}$ ,  $\hat{p}_{FSW}$ , and three average scores  $\hat{s}_{DF}$ ,  $\hat{s}_{F2F}$ ,  $\hat{s}_{FSW}$ . Then, the three estimated labels are passed to a fusion block that applies the logical OR operator (indicated as

$\vee$ ) in order to get  $\hat{p}$ . In other words, a video is classified as manipulated as soon as one of the three detectors returns the label 1. Furthermore, in case of  $\hat{p} = 1$ , the maximum value of the scores is selected as indicator of the manipulation technique used to create the video.

Table IV reports the accuracy results obtained through this approach for the two variants selected in Section V-A,  $(F, \rightarrow)$  and  $(B, \leftrightarrow)$ , which consistently exceed 85%. We also report the false positive rate (fraction of original videos erroneously classified as manipulated) and the false negative rate (fraction of manipulated videos erroneously classified as original). The former seems to be more crucial for this fused approach, likely due to the fact that original videos are underrepresented in the overall training set.

Finally, we measure the accuracy in estimating the manipulation technique used when a video is correctly classified as manipulated. Table V and Table VI are the confusion matrices of the two classifiers for this task. The high diagonal values (around 90,00% in most cases) indicate that the feature representation carries quite strong information on the specific manipulations techniques.

### C. Impact of Strong Video Compression

The FaceForensics++ dataset also offers a more heavily compressed version of the videos, i.e., with  $cf = 40$ . As reported in [40], the quality degradation due to compression compromises the performance of detection algorithms, as well as humans. We therefore assess how this impacts our method by reproducing the single-technique scenario for the two best performing classifiers, and report the results in Figure 8 and Table VII. While keeping an average accuracy around 70%, the performance decrease is evident when compared to Figure 6 (around 20%), thus confirming that, as most of the existing methods, our feature representation also suffers from the

TABLE IV: Classification results in the multiple-technique scenario.

Algorithm Version	False Positive Rate	False Negative Rate	Accuracy
$(F, \rightarrow)$	20,00%	<b>11,43%</b>	86,43%
$(B, \leftrightarrow)$	15,71%	11,90%	<b>87,14%</b>

TABLE V: Confusion matrix for the manipulation estimation task with  $(F, \rightarrow)$ .

		PREDICTIONS		
		DF	F2F	FSW
TARGET	DF	<b>89,23</b>	10,77%	0,00%
	F2F	5,17%	<b>91,38%</b>	3,45%
	FSW	0,00%	3,17%	<b>96,83%</b>

TABLE VI: Confusion matrix for the manipulation estimation task with  $(B, \leftrightarrow)$ .

		PREDICTIONS		
		DF	F2F	FSW
TARGET	DF	<b>83,33</b>	16,17%	0,00%
	F2F	5,08%	<b>91,53%</b>	3,39%
	FSW	0,00%	5,00%	<b>95,00%</b>

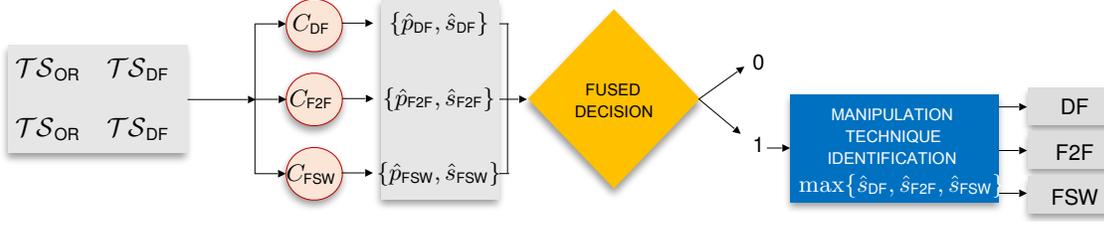


Fig. 7: Decision pipeline for the multiple-technique scenario.

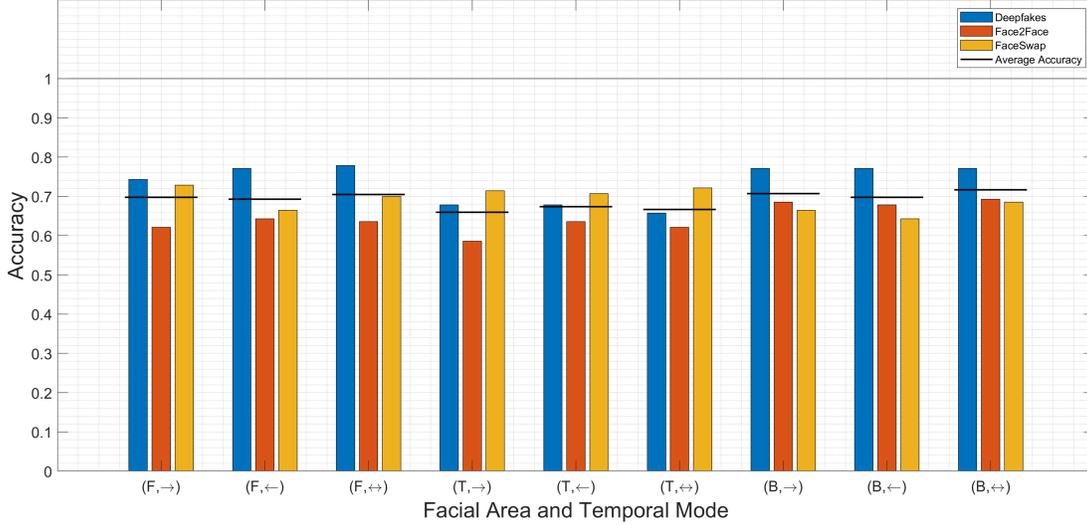


Fig. 8: Classification accuracy per manipulation technique in case of strong video compression.

TABLE VII: Classification accuracy and AUC computed on the single-manipulation scenario in case of strong video compression.

Algorithm Version	Accuracy			Average Accuracy	AUC			Average AUC
	Deepfakes	Face2Face	FaceSwap	Cross-Dataset	Deepfakes	Face2Face	FaceSwap	Cross-Dataset
(F, $\rightarrow$ )	74,29%	62,14%	72,86%	69,76%	80,49%	68,97%	80,59%	76,68%
(B, $\leftrightarrow$ )	77,14%	69,29%	68,57%	<b>71,67%</b>	81,35%	74,04%	79,64%	<b>78,34%</b>

TABLE VIII: Classification accuracy in the multiple-technique scenario in case of strong video compression.

Algorithm Version	Decision Criterion	False Positive Rate	False Negative Rate	Accuracy
(F, $\rightarrow$ )	$\checkmark$	50,00%	27,62%	66,79%
(B, $\leftrightarrow$ )	$\checkmark$	42,86%	<b>23,81%</b>	<b>71,43%</b>

application of a heavier compression. This holds also for the multiple-technique scenario, where the accuracy of the best classifier drop to 71% (see Table VIII).

#### D. Comparison with other descriptors

In this subsection, we consider the performance of our method with respect to other detection algorithms.

We first compare our feature representation with a known competitor among the spatio-temporal texture descriptors used in face anti-spoofing, the LBP-TOP [54]. Differently from LDPs, LBPs capture only information on the first-order directional

derivatives computed at a central reference pixel, that are thresholded, encoded into a binary number, and finally collected into histogram over different pixels; LBP-TOP is the corresponding temporal extension and yields a feature vector of length  $[1, 177]$ , obtained by applying the uniform pattern version of the LBP features that led to a more compact feature vector and descriptor robust to rotations. We want to determine whether and how much the improved performance observed in [38] for the face spoofing detection task generalizes to the detection of facial manipulations. To this purpose, the tests performed in Section V-A are extended by replacing the LDP-TOP feature vector with the LBP-TOP one, while keeping unchanged all the other steps described in Sections III and IV. We report in Fig. 9 the classification accuracy loss observed when using LBP-TOP instead of LDP-TOP (i.e., with respect to the results in Figure 6). The loss is always positive, thus LDP-TOP indeed outperforms LBP-TOP by a significant margin.

Then, we position our results with respect to other methods proposed in literature and benchmarked on the same dataset in [40]. Since the training, validation, and testing splits of the

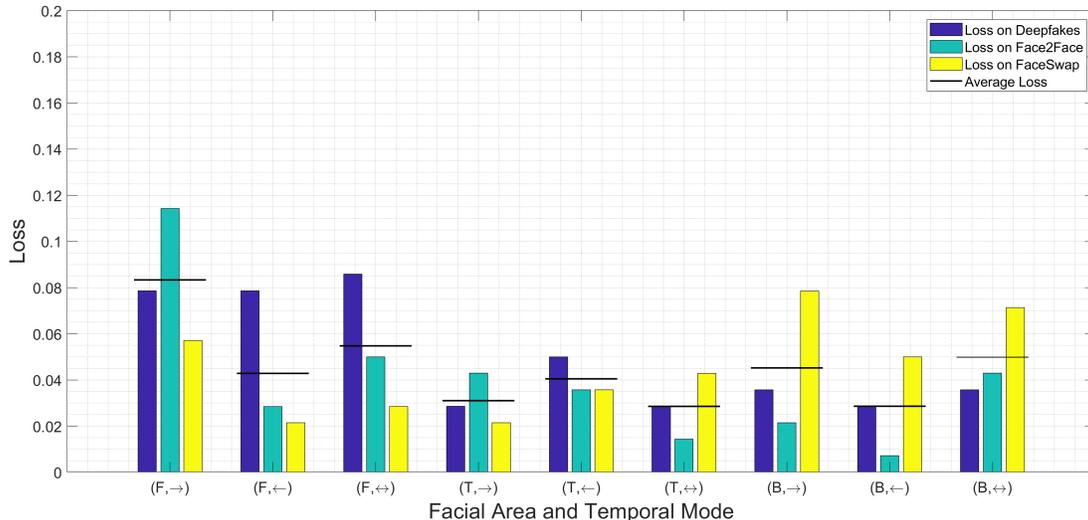


Fig. 9: Classification accuracy loss per manipulation technique when using LBP-TOP descriptors instead of the proposed ones.

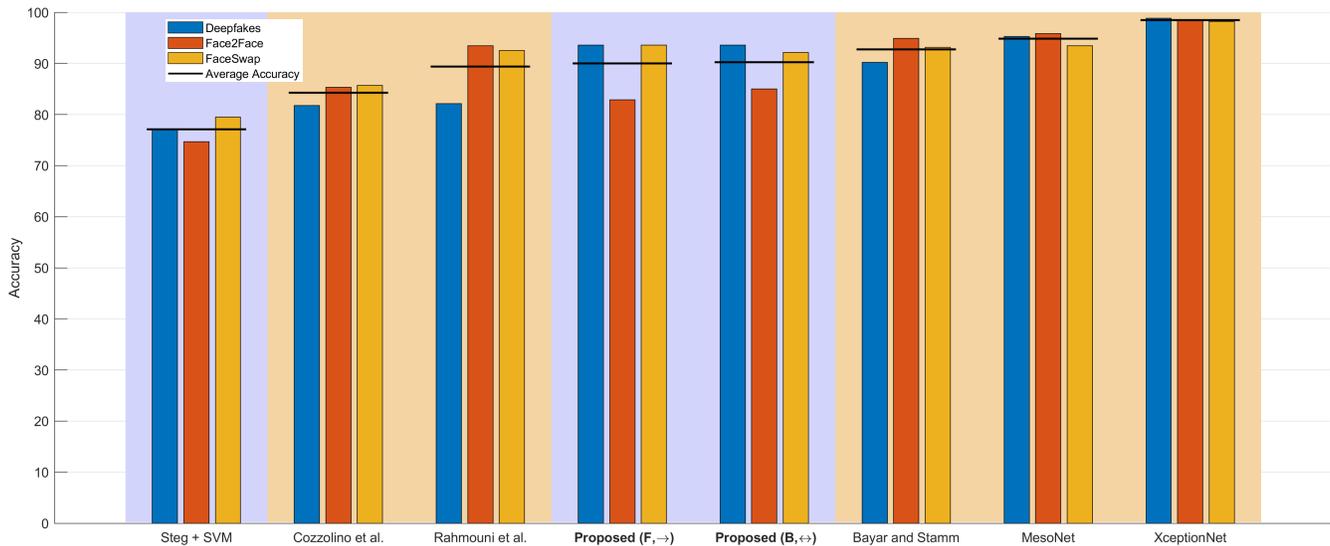


Fig. 10: Classification accuracies of the proposed algorithms with respect to other detection methods. Orange background indicates that the method is based on CNNs.

FaceForensics++ dataset are standard, it is fair to compare the results obtained through our proposed pipelines with the ones reported in [40] in terms of accuracy on the testing set. Figure 10 reports the results of our  $(F, \rightarrow)$  and  $(B, \leftrightarrow)$  classifiers and other six detection methods, namely “Steg+SVM” [20], “Cozzolino et al.” [14], “Rahmouni et al.” [39], “Bayar and Stamm” [7], “MesoNet” [3], and “XceptionNet” [40], sorted according to their average accuracy over manipulation techniques. All of them, except for the “Steg+SVM”, are based on convolutional neural networks. Remarkably, our approach outperforms the SVM-based one [20] by a large margin, and also two techniques based on CNNs [14] and [39]. While the performance of other deep networks like XceptionNet remains significantly better,

the proposed spatio-temporal descriptors, separated linearly in the feature space, provide fairly accurate results with the advantages of higher explainability of the encoded patterns and limited training time.

## VI. CONCLUSIONS

In this paper we have proposed a novel methodology to detect fake video sequences by exploiting spatio-temporal descriptors successfully exploited for the task of face anti-spoofing. Results show good performance on various manipulation techniques and in different experimental scenarios. Relatively small feature representation and relatively simple

classifiers allow to detect manipulated video sequences and identify the adopted manipulation technique.

Future work will deal with the challenging problem of heavy video compression, where current literature still does not achieve satisfactory results. Moreover, further extension will consider the scenario where new manipulation techniques could be considered and learned by the detector, e.g. by exploiting innovative paradigms coming from the machine learning domain like incremental learning.

#### ACKNOWLEDGMENTS

This work was supported by the project PREMIER (PRE-serving Media trustworthiness in the artificial Intelligence ERA), funded by the Italian Ministry of Education, University, and Research (MIUR) within the PRIN 2017 program. The second author was partially supported by Archimedes Privatstiftung, Innsbruck.

#### REFERENCES

- [1] Deepfakes Github. <https://github.com/deepfakes/faceswap>, 2019.
- [2] Faceswap. <https://github.com/MarekKowalski/FaceSwap/>, 2019.
- [3] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen. Mesonet: a compact facial video forgery detection network. In *IEEE International Workshop on Information Forensics and Security*, pages 1–7, 2018.
- [4] S. Agarwal, H. Farid, Y. Gu, M. He, K. Nagano, and H. Li. Protecting world leaders against deep fakes. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.
- [5] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo. Deepfake video detection through optical flow based cnn. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2019.
- [6] Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, and Jianzhuang Liu. Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor. *IEEE Transactions on Image Processing*, pages 533–544, 2010.
- [7] B. Bayar and M. C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018.
- [8] M. Bonomi and G. Boato. Digital human face detection in video sequences via a physiological signal analysis. *Journal of Electronic Imaging*, 29(1):1 – 10, 2020.
- [9] D. Chen, J. Li, S. Wang, and S. Li. Identifying computer generated and digital camera images using fractional lower order moments. In *2009 4th IEEE Conference on Industrial Electronics and Applications*, pages 230–235, 2009.
- [10] U. A. Ciftci and I. Demir. Fakecatcher: Detection of synthetic portrait videos using biological signals. *CoRR*, 2019.
- [11] V. Conotter, E. Bodnari, G. Boato, and H. Farid. Physiologically-based detection of computer generated faces in video. In *IEEE International Conference on Image Processing (ICIP)*, 2014.
- [12] D. Dang-Nguyen, G. Boato, and F. G. B. D. Natale. Discrimination between computer generated and natural human faces based on asymmetry information. In *European Signal Processing Conference (EUSIPCO)*, pages 1234–1238, 2012.
- [13] D.-T. Dang-Nguyen, G. Boato, and F. G. B. De Natale. 3d-model-based video analysis for computer generated faces identification. *IEEE Transactions on Information Forensics and Security*, pages 746–761, 2015.
- [14] G. P. Davide Cozzolino and L. Verdoliva. Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection. In *ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec)*, 2017.
- [15] A. E. Dirik, H. T. Sencar, and N. Memon. Source camera identification based on sensor dust characteristics. In *IEEE Workshop on Signal Processing Applications for Public Security and Forensics*, pages 1–6, 2007.
- [16] R. Durall Lopez, M. Keuper, F.-J. Pfreundt, and J. Keuper. Unmasking deepfakes with simple features. *CoRR*, 2019.
- [17] Facebook. Deepfake Detection Challenge (DFDC). <https://deepfakedetectionchallenge.ai>, 2020.
- [18] S. Fernandes, S. Raj, E. Ortiz, I. Vintila, M. Salter, G. Urosevic, and S. Jha. Predicting heart rate variations of deepfake videos using neural ode. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.
- [19] T. d. Freitas Pereira, J. Komulainen, A. Anjos, J. M. De Martino, A. Hadid, M. Pietikäinen, and S. Marcel. Face liveness detection using dynamic texture. *EURASIP Journal on Image and Video Processing*, 2014.
- [20] J. Fridrich and J. Kodovsky. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, pages 868–882, 2012.
- [21] A. C. Gallagher and T. Chen. Image authentication by detecting traces of demosaicing. In *IEEE Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2008.
- [22] D. Güera and E. J. Delp. Deepfake video detection using recurrent neural networks. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018.
- [23] H. R. Hasan and K. Salah. Combating deepfake videos using blockchain and smart contracts. *IEEE Access*, pages 41596–41606, 2019.
- [24] T. Jabid, M. Kabir, and O. Chae. Local directional pattern (ldp) for face recognition. *International Journal of Innovative Computing, Information and Control*, pages 329–330, 2010.
- [25] Y. Ke, W. Min, X. Du, and Z. Chen. Detecting the composite of photographic image and computer generated image combining with color, texture and shape feature. *Journal of Theoretical and Applied Information Technology*, pages 844–851, 2013.
- [26] A. Khodabakhsh, R. Ramachandra, K. Raja, P. Wasnik, and C. Busch. Fake face detection methods: Can they be generalized? In *International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–6, 2018.
- [27] Y. Li, M. Chang, and S. Lyu. In icu oculi: Exposing ai created fake videos by detecting eye blinking. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, 2018.
- [28] Y. Li, M. Chang, and S. Lyu. In icu oculi: Exposing AI created fake videos by detecting eye blinking. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–7, 2018.
- [29] S. Lyu and H. Farid. How realistic is photorealistic? *IEEE Transactions on Signal Processing*, pages 845–850, 2005.
- [30] F. Marra, D. Gragnaniello, D. Cozzolino, and L. Verdoliva. Detection of gan-generated fake images over social networks. In *IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 384–389, 2018.
- [31] F. Marra, C. Saltori, G. Boato, and L. Verdoliva. Incremental learning for the detection and classification of GAN-generated images. In *IEEE Workshop on Information Forensics and Security (WIFS)*, 2019.
- [32] F. Matern, C. Riess, and M. Stamminger. Exploiting visual artifacts to expose deepfakes and face manipulations. In *IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pages 83–92, 2019.
- [33] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie, and M.-P. Tsui. Physics-motivated features for distinguishing photographic images and thecomputer graphics. In *ACM International Conference on Multimedia*, pages 239–248, 2005.
- [34] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen. Multi-task learning for detecting and segmenting manipulated facial images and videos. *CoRR*, abs/1906.06876, 2019.
- [35] T. T. Nguyen, C. M. Nguyen, D. T. Nguyen, D. T. Nguyen, and S. Nahavandi. Deep learning for deepfakes creation and detection. *ArXiv*, 2019.
- [36] F. Pan, J. Chen, and J. Huang. Discriminating between photorealistic computer graphics and natural images using fractal geometry. *Science in China Series F: Information Sciences*, pages 329–337, 2009.
- [37] F. Peng, D.-L. Zhou, L. Min, and X.-M. Sun. Discrimination of natural images and computer generated graphics based on multi-fractal and regression analysis. *AEU - International Journal of Electronics and Communications*, 2016.
- [38] Q.-T. Phan, D.-T. Dang-Nguyen, G. Boato, and F. G. B. D. Natale. Face spoofing detection using ldp-top. *2016 IEEE International Conference on Image Processing (ICIP)*, pages 404–408, 2016.
- [39] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen. Distinguishing computer graphics from natural images using convolution neural networks. In *IEEE Workshop on Information Forensics and Security (WIFS)*, pages 1–6, 2017.
- [40] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. In *IEEE International Conference on Computer Vision*, 2019.
- [41] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 80–87, 2019.

- [42] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, pages 1627–1639, 1964.
- [43] L. Souza, L. Oliveira, M. Pamplona, and J. Papa. How far did we get in face spoofing detection? *Engineering Applications of Artificial Intelligence*, 72(C):368–381, 2018.
- [44] J. Stehouwer, H. Dang, F. Liu, X. Liu, and A. K. Jain. On the detection of digital face manipulation. *CoRR*, abs/1910.01717, 2019.
- [45] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. *Commun. ACM*, pages 96–104, 2018.
- [46] L. Verdoliva. Media forensics and deepfakes: an overview. *ArXiv*, abs/2001.06564, 2020.
- [47] J. Vincent. Watch Jordan Peele use AI to make Barack Obama deliver a PSA about fake news, 2019.
- [48] H.-Y. Wu, M. Rubinstein, E. Shih, J. Guttag, F. Durand, and W. Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM Transactions on Graphics*, pages 65:1–65:8, 2012.
- [49] J. Xiao, S. Li, and Q. Xu. Video-based evidence analysis and extraction in digital forensic investigation. *IEEE Access*, pages 55432–55442, 2019.
- [50] X. Yang, Y. Li, and S. Lyu. Exposing deep fakes using inconsistent head poses. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265, 2019.
- [51] X. Yang, Y. Li, H. Qi, and S. Lyu. Exposing gan-synthesized faces using landmark locations. In *ACM Workshop on Information Hiding and Multimedia Security (IH&MMsec)*, pages 113–118, 2019.
- [52] N. Yu, L. Davis, and M. Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 7555–7565, 2019.
- [53] X. Zhang, S. Karaman, and S.-F. Chang. Detecting and simulating artifacts in GAN fake images. In *IEEE Workshop on Information Forensics and Security (WIFS)*, 2019.
- [54] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.*, (6):915–928, 2007.