A Comparative Evaluation of Local Feature Descriptors for DeepFakes Detection

Zahid Akhtar
Department of Computer Science
University of Memphis
Memphis, USA
zmomin@memphis.edu

Dipankar Dasgupta
Department of Computer Science
University of Memphis
Memphis, USA
ddasgupt@memphis.edu

***Abstract*—The global proliferation of affordable photographing devices and readily-available face image and video editing software has caused a remarkable rise in face manipulations, e.g., altering face skin color using FaceApp. Such synthetic manipulations are becoming a very perilous problem, as altered faces not only can fool human experts but also have detrimental consequences on automated face identification systems (AFIS). Thus, it is vital to formulate techniques to improve the robustness of AFIS against digital face manipulations. The most prominent countermeasure is face manipulation detection, which aims at discriminating genuine samples from manipulated ones. Over the years, analysis of microtextural features using local image descriptors has been successfully used in various applications owing to their flexibility, computational simplicity, and performances. Therefore, in this paper, we study the possibility of identifying manipulated faces via local feature descriptors. The comparative experimental investigation of ten local feature descriptors on a new and publicly available DeepfakeTIMIT database is reported.**

***Index Terms*—Face recognition, Deepfake video, Face manipulation, Person Identification**

I. Introduction

Human face has affluent characteristics that provide powerful biometric cues to recognize people. In fact, face recognition systems are extensively being adopted in different applications spanning from law enforcement, border control systems to personal authentication on smartphone [1]. Face has become a natural choice for recognition tasks as it is not hidden, and it conveys identity in social meetings as well as in digital communications. Moreover, face identification not only works exceptionally well in unconstrained environments but also does not require physical contact or advanced hardware [2].

Nevertheless, large number of people are utilizing easily available tools (e.g., FaceApp, Face2Face, Deepfake, AgingBooth, Adobe Photoshop and PotraitPro Studio) to digitally edit face images or videos for innocuous/recreational (e.g., face sample retouching to look more attractive) or spiteful (e.g., propagating fake news) motives. Such synthetic manipulations have detrimental effects on face recognition algorithms. Manipulations usually alter the face appearances so much so that they negatively affect automated face identification system (AFIS) [35]. For example, the studies by Ferrara *et al.* [3] and Korshunov *et al.* [4] showed that face morphing could increase the error rates of AFIS by 50%. Likewise, different commercial AFISs' recognition accuracy drops around 40% when faces are altered by make-ups [5], whereas 60-80% under face spoofing [6]. More than 180 countries have employed face verification at the border control; thus, face image manipulation is extremely critical challenge in such scenarios that use electronic passports (ePass). Many countries issue or renew the ePass merely on the basis of photo submitted online by applicant. Thus, a person with vested interest, for instance, may morph their face image with another person such that provided face image in ePass can be matched to both persons [7]. Likewise, people can manipulate their face samples to look older or younger in order to deceive age-specific access control.

There exist different works in the scientific literature on face attribute editing. For instance, Persch *et al.* [8] developed a method to changing skin color (i.e., ethnicity) of a face by morphing texture descriptors guided by a different source image. Lately, advances in deep learning have made it possible to seamlessly manipulate faces that looks exceedingly realistic. For example, Korshunova *et al.* [9] designed multi-scale appearance-based convolution neural networks (CNNs) to swap faces in images and videos, also are known as "DeepFakes". Quintessential countermeasure to face manipulation is face authenticity detection methods, which aim at disambiguating human benign face samples from digitally manipulated ones [4]. For example, Seibold *et al.* [10] used well-known deep architectures (e.g., AlexNet) to identify face morphing. Despite the recent progress, sophisticated face manipulations are still very difficult to be detected by human experts and present image forensics tools [4].

In this paper, we focus on the specific issue of face swapping detection. In particular, this paper aims at comparatively analyzing the potential of image local feature descriptors for face swapping detection. Most widely adopted ten local facial representations namely LBP (Local Binary Patterns) [11], FDLBP (Frequency Decoded Local Binary Pattern) [12], QLRBP (Quaternionic Local Ranking Binary Pattern) [13], BGP (Binary Gabor Pattern) [14], LPQ (Local Phase Quantization) [15], BSIF (Binarized Statistical Image Features) [16], CENTRIST (CENsus TRansform hISTogram) [17], PHOG (Pyramid Histogram of Oriented Gradients) [18], SIFT (Scale invariant feature transform) [19] and SURF (speeded up robust

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

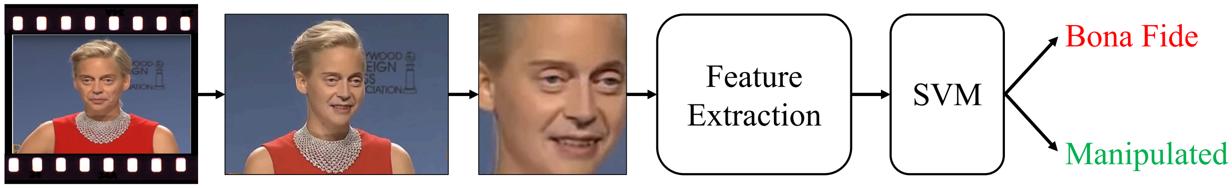

Fig. 1. Schematic of the procedure of performance evaluation for the DeepFake detection using image local descriptors.

features) [20] have been considered in this study. Experiments have been conducted on a newly collected DeepfakeTIMIT database.

The remainder of this paper is organized as follows. Section II presents prior works on face authenticity. The process of face manipulation detection together with ten facial local descriptors is briefly described in Section III. Experiments are described and discussed in Section IV, while conclusions are drawn in Section V.

II. Related Work

Sophisticated face editing can not only fool humans but also AFISs. In particular, face swapping (i.e., deepfake) is a momentous challenge, as fake information with manipulated multimedia can swiftly circulate via social networks, which may lead to troublesome effects. For instance, a person can modify the content and people in a video to spread fake news, which may lead to war between two nations.

Yang *et al.* [22] developed a method for face expression editing via geometry of the input as an iterative optimization. Whereas, one of the seminal works to swap faces was carried out by Bitouk *et al.* [23]. Authors aimed to de-identify the input face to protect privacy by blending two similar looking faces. Since the aim was de-identification, thus the swapped faces produced were not seamless. In turn, several algorithms have been proposed for face swapping. For example, Arar *et al.* [24] presented a scheme using Active Appearance Model (AAM). Xingjie *et al.* [25] designed a framework to swap the faces using alpha image blending method. Whereas, den Uyl *et al.* [26] designed AAM with adaptive texture sampling technique. Mahajan *et al.* [27] developed a technique that employed facial landmarks and blending of feature, and Nirkin *et al.* [28] presented a 3D face shape swapping using bilinear interpolation. A number of face swapping approaching using DNNs have been formulated. For instance, Natsume *et al.* [29] proposed FSNet that is a generative model. While, Suwajanakorn *et al.* [30] utilized LSTM (Long Short-Term Memory) to create an architecture that can synthesize mouth features only from an audio speech. Inspired by the success of DNNs based face manipulation schemes, not only open source but also commercial apps can be seen everywhere [27].

The rapid growth in effortlessly generating deepfake images and videos is demanding a dependable detection scheme. Comparatively, there are very limited number of publications as well as databases for identifying and evaluating manipulated faces. For instance, Zhang *et al.* [31] developed a technique based on bag of words to recognize samples with face substitutions. Agarwal *et al.* [32] applied weighted local magnitude patterns for swapped faces via Snapchat. Afchar *et al.* [33] presented an automatic traditional CNNs based detection method for face tampering in videos generated via Deepfake and Face2Face apps. Whereas, Li *et al.* [34] demonstrated that fake face videos can be detected by combining CNNs with RNN to analyze the eye blinking.

III. Local Feature Descriptors based DeepFakes Detection Systems

Face DeepFakes detection is normally considered as a binary classification issue, where input video/image has to be flagged as either manipulated or benign. The keynote of the procedure is procuring a distinctive feature set that when used with a classification scheme gives the likelihood of the input sample realism. Fig. 1 depicts the overall schematic diagram for performance evaluation of the DeepFake detection using local image descriptors. The framework first extracts an individual frame from a given input face video. Then, the face is detected using Viola-Jones algorithm. Next, the face is normalized, and features are extracted via a specific local image descriptor among the ones considered in this study. The extracted features are fed to an SVM classifier to acquire the final binary decision: manipulated (DeepFake) or bona fide.

A. Local Feature Descriptors

Here, we describe image local descriptors evaluated in this paper for face authenticity detection.

1) Local Binary Pattern (LBP): Ahonen *et al.* [11] proposed the original LBP to describe the spatial structure of a local image textures. The LBP operator encodes the pixels of an image by thresholding the neighborhood of each pixel with the center pixel's value and considering the result as a binary number. The binary string is used to obtain the histogram as a feature set. The LBP code of center pixel (x, y) can be defined as:

$$\begin{gathered}\mathrm{LBP}_{P,R}(x,y)=\sum_{p=0}^{p-1}s(g_p-g_c)2^p\\ s(z)=\begin{cases}1, z\geq 0,\\ 0, z<0\end{cases}\end{gathered}\tag{1}$$

where P, R and $g_c, g_0, \ldots, g_{P-1}$ are, respectively, the number of sampling points, a circle of radius, the center pixel (x, y)'s gray value and the pixels gray values in the circular neighborhood.

2) Frequency Decoded Local Binary Pattern (FDLBP): Dubey *et al.* [12] developed a face representation called FDLBP based on frequency decoders. The steps of FDLBP can be summarized as: i) image filtering–the input image is first processed using one low-pass filter (i.e., average filter for coarse information) and four high-pass filters (i.e., horizontal-vertical difference, diagonal difference, Sobel vertical edge and Sobel horizontal edge filters for detained information); ii) LBP binary code generation–LBP codes are obtained by employing LBP operator on filtered images; iii) frequency decoders–LBP binary codes are given to two encoders in order to use the relationship among different frequency filtered images; iv) feature vector computation–a standard histogram strategy is applied on the output of decoders to generate FDLBP feature vector.

3) Quaternionic Local Ranking Binary Pattern (QLRBP): Lan *et al.* [13] formulated QLRBP as a color texture feature descriptor depending on Quaternion, which is a complex number containing three imaginary and one real parts. In QLRBP, the imaginary components are utilized to represent a color pixel in an image. Specifically, a window of 3×3 neighborhood is applied, and a reference color pixel $I^{'}(r^{'}, g^{'}, b^{'})$ and a color pixel $I(r, g, b)$ is employed to drive QLRBP operator. In the window, he CTQ (Clifford Translation of Quaternionic) that is complex together with a rank-based LBP technique are utilized to encode and rank the color pixels. The phase of the function of the function is used for ranking. The CTQ between two color pixels is given by:

$$\theta(x,y) = \tanh^{-1} \frac{\sqrt{(gb' - bg')^2 + (br' - rb')^2 + (rg' - gr')^2}}{-(rr' + gg' + bb')} \tag{2}$$

where $r = r(x, y), g = g(x, y)$ and $b = b(x, y)$. The angles are computed for all 9 pixels in the window. The phase angle $\theta(x, y)$ (and corresponding QLRBP) is between two vectors I and $I^{'}$. To differential the weights to individual color channels, the weighted L_1 Phase is employed as:

$$\theta(x,y) = \tanh^{-1} \frac{\alpha_1|(gb^{'} - bg^{'})| + \alpha_2|(br^{'} - rb^{'})| + \alpha_3|(rg^{'} - gr^{'})|}{-(rr' + gg' + bb')} \tag{3}$$

By setting $I^{'}$ to a certain value, the final features are estimated.

4) Binary Gabor Pattern (BGP): Zhang *et al.* [14] designed BGP to integrate the advantages of both Gabor filters and LBP. To make features robust against noise, BGP utilizes difference between image regions instead of different between two pixels as in LBP. The difference between regions is computed using Gabor filters. Specifically, BGP feature descriptor is computed by first convolving in the image with a filter bank (i.e., 8 different orientations Gabor filters). Resultant convoluted responses are thresholded depending on sign to achieve binarized vector (B). Then, the n-binary representation at pixel is converted into an integer number (i.e., rotation sensitive BGP) using n-orientations as $BGP^{'} = \sum_{i=0}^{7} B_i 2^i$. To address rotation invariance BGP, BGP' values are grouped together as:

$$BGP_{ri} = max(ROR(BGP, i)|i = 0 \ldots 7) \tag{4}$$

where ROR performs a circular bitwise right shift on the n-bit number.

5) Local Phase Quantization (LPQ): Ojansivu *et al.* [15] presented LPQ that is robust to blur and low resolution as it is extracted by quantizing the Fourier transform phase in local neighborhoods. The pixel is analyzed using short-time Fourier transform (STFT):

$$\hat{I}_x = \sum_y I(y) w(y - x) e^{-j2\pi u \cdot y} \tag{5}$$

where (x, y), $u, w(.)$ and $\hat{I}_x(.)$ are spatial coordinates, spatial bi-directional frequencies, a compact window, and output STFT around x. Next, only four frequencies to direction $0°$ (i.e., $u_0 = (\alpha, 0)$), $45°$ (i.e., $u_1 = (\alpha, \alpha)$), $90°$ (i.e., $u_2 = (0, \alpha)$), and $135°$ (i.e., $u_3 = (-\alpha, \alpha)$ are considered. The LPQ basic features are nothing but phases of the chosen coefficients as $F_i(x) = \angle \hat{I}_x(u_i)$. The features are quantized with a quantizer in $[-\pi, \pi]$ to provide indexes C_i(x), which is finally summarized in a scalar values between 0 to 255: C(x) = LPQ(x) = $\sum_{i=0}^{3} C_i(x) 4^i$.

6) Binarized statistical image features (BSIF): Kannala *et al.* [16] constructed BSIF with the binarization of the responses to linear filters that are learnt from natural images and ICA (independent component analysis). The learnt filters are utilized to encode every pixel as binary string, which is used to generate histogram as BSIF descriptor. There are three steps for BSIF, i.e., mean subtraction of patches, reducing dimensionality via PCA, and computing statistically independent filters with ICA. For a face image I and a filter F_i of size $l \times m$, the convoluted response is obtained as $r_i = \sum_{l,m} I(l, m) F_i(l, m)$, where $F_i \forall = \{1, \ldots, m\}$ depicts filters that is employed to attain binary string as $b_i = 1$ if $r_i > 0$, otherwise $b_i = 0$. The BSIF descriptor is eventually estimated as a normalized histogram of binary codes.

7) CENsus TRansform hISTogram (CENTRIST): CENTRIST [17] depends on Census Transform (CT) that compares the pixel value with its eight neighboring pixels such that if the center pixel's intensity value is bigger than (or equal to) it's given neighbor then a bit 1 is set in the respective position, otherwise 0. The 8-bit binary sting for every pixel is then transformed in a base-10 number between 0 to 255. Finally, all CT values are converted into a histogram known as CENTRIST descriptor. CENTRIST is robust to gamma and illumination changes.

8) Pyramid of Histogram of Oriented Gradients (PHOG): PHOG encodes face image utilizing its local shape at various scales with the help of distribution of direction of intensity and edges [18]. PHOG is identical to HOG, but in a pyramid hierarchy. First Canny edge detector is applied to the image, which is then tessellated in smaller cells depending on numbers of

978-1-7281-5092-5/19/$31.00 ©2019 IEEE

levels. The histograms of gradient orientations' occurrences in these cells produce PHOG descriptor.

9) Scale Invariant Feature Transform (SIFT): It was developed by Lowe, 2004 to obtain features that are mainly invariant to scaling, rotation and rotation. It has four steps: i) scale space extreme detection (i.e., scanning the image together with computing maxima of the Difference of Gaussian (DoG)); ii) keypoint localization (i.e., using Hessian matrix to retain potential interest points with higher contrast and edge information); iii) orientation assignment (i.e., forming orientation histogram of each DoG local maximum around the keypoints); (4) keypoint descriptor (i.e., constructing a feature vector by considering gradient strength indicating keypoint direction). Specifically, the selected region is tessellated into 4×4 sub-regions with 8 orientation bins to yield $4 \times 4 \times 8 = 128$ elements as SIFT features.

10) Speeded Up Robust Features (SURF): Bay *et al.* [20] developed SURF as an improvement of SIFT features. SURF is computed using 2D Haar wavelet and integral images. The 2D Haar wavelet is secured by applying Hessian matrix to extract blob-like structures at positions where determinant is maximum. Namely, region around each keypoint is divided in 4×4 to compute 2D Haar wavelet response with the help of integral images. Since every response gives 4 values, every keypoint is described with a $4 \times 4 \times 4 = 64$-dimensional feature vector, which is scale and rotation invariant.

TABLE I
COMPARISON OF LOCAL FEATURE DESCRIPTORS FOR FACE AUTHENTICITY DETECTION IN TERMS OF EER (%) AND FRR@FAR10% (%).

Method	**EER (%)**	**FRR@FAR10%(%)**
Low quality face manipulation (DeepFake) data subset		
LSTM lip-sync [21]	41.80	81.67
IQM + SVM [4]	3.33	0.95
LPB	8.01	6.07
FDLBP	21.58	54.53
QLRBP	8.98	8.45
BGP	2.46	0.60
LPQ	2.76	0.73
BSIF	24.62	41.36
CENTRIST	7.89	6.05
PHOG	83.14	100
SIFT	57.10	94.37
SURF	45.46	78.23
High quality face manipulation (DeepFake) data subset		
IQM + SVM [4]	8.97	9.05
LPB	17.16	43.02
FDLBP	37.19	88.30
QLRBP	27.70	59.49
BGP	13.33	15.99
LPQ	13.69	16.53
BSIF	60.88	93.69
CENTRIST	11.43	13.12
PHOG	89.70	100
SIFT	57.58	95.43
SURF	67.26	98.17

IV. EXPERIMENTS

Here, we provide a comparative experimental evaluation of the considered local feature descriptors.

A. Dataset

We used recently released public DeepfakeTIMIT database [4]. For both genuine and tampered subsets, 10 videos per subject is available. In this study, 320 genuine and 640 manipulated videos from 32 subjects were used. The genuine videos are from VidTIMIT database (http://conradsanderson.id.au/vidtimit/) that were captured in controlled environment. Deepfake/manipulated videos were generated using GAN-based approach (https://github.com/shaoanlu/faceswap-GAN) to swap the faces from source to target with two qualities, i.e., low and high.

B. Experimental Protocol

To evaluate the efficacy of local feature descriptors for face manipulation (DeepFake) detection, all videos were split into train and test subsets. Both train and test subsets were composed of samples (genuine and manipulated) from 50% of all users. In order to avoid bias in train and test phases, the users were arranged such that the same user/subject did not appear in both train and test subsets. Also, each individual frame of every video was considered as a distinct sample. For feature extraction, the detected face image is first normalized to 200×200 pixels. A face authenticity detection technique is subject to two types of errors, i.e., False Acceptance Rate (FAR) and False Rejection Rate (FRR). The FAR is portion of manipulated samples incorrectly classified as bona fide. The FRR is portion of bona fide samples as manipulated. We evaluated performance using Equal Error Rate (EER, i.e. FAR = FRR) and FRR (by employing the threshold when FAR = 10%).

C. Experimental Results

In Table I, we report the performance comparison between the adopted individual local image descriptors along with the existing methods to discriminate manipulated (DeepFake) from genuine face samples in terms of EER and FRR@FAR10%, when face manipulations were of low and high quality. Several observations can be extracted from Table I. For example, BGP all in all outperforms other considered local descriptors, e.g., under low quality face swapping, BGP attained 2.46% EER, whereas 24.62% and 45.46% by BSIF and SURF, respectively. Similarly, under high quality face swapping, EERs for BGP and LBP are 13.33% and 17.16%, respectively. The superior performance is largely because BGP is computed using differences between regions, instead of pixels, at different resolutions, scales, and gradient magnitudes. Moreover, it is robust to noise and rotation. LPQ achieved similar accuracy due to use of Gabor filters like in BGP. For instance, for high quality face manipulations, LPQ procured 16.53% FRR@FAR10%, while 15.99% by BGP.

Face authenticity detection schemes' proficiency depends on manipulation quality. Low quality DeepFakes are easier to be distinguished by local feature descriptors, while high quality ones could be detected better by quality related features such as the method proposed by Korshunov *et al.* [4]. Specifically, the EERs for lower and higher quality samples attained by LBP, respectively, are 8.01% and 17.16%, whereas by IQM technique [4] are 3.33% and 8.97%. By metaknowledge analysis, we can state that high quality DeepFakes are very hard to be detected by not only humans but also local descriptors and existing method using temporal or image quality features.

We can also observe in Table I that all assessed descriptors (except PHOG) in this study attained superior performance than LSTM lip-sync [21]. For example, under low quality manipulations, the FRR@FAR10% reached by LSTM lip-sync, QLRBP and PHOG, respectively, are 81.67%, 8.45% and 100%. Besides, the framework presented in this work is simple and has low computational complexity, since it does require mouth region detection as in the study by Korshunov *et al.* [21], which makes presented framework more suitable for mobile platforms. Moreover, local features that were used for face authenticity detection may also be applied for face recognition.

V. Conclusion

This study evaluated the performance of ten local feature descriptors for the task of face manipulation detection. The experiments were conducted on a publicly available DeepfakeTIMIT database, which is composed of Deepfake face samples with low- and high-quality face swapping obtained by GAN-based technique. Most image local descriptors in this work are efficient in identifying low quality manipulated faces, while their performance, contrary to image related method, degrades when high quality manipulation samples are observed. Future works includes extending the evaluation on different sophisticated manipulations datasets.

References

[1] Z. Akhtar, A. Rattani, A. Hadid, M. Tistarelli, “Face recognition under ageing effect: a comparative analysis”, International Conference on Image Analysis and Processing, pp. 309-318, 2013.

[2] Z. Akhtar and A. Rattani, “A face in any form: new challenges and opportunities for face recognition technology”, IEEE Computer, vol. 50, no. 4, pp. 80-90, 2017.

[3] M. Ferrara, A. Franco, D. Maltoni, “On the effects of image alterations on face recognition accuracy”, Face recognition across the imaging Spectrum, Springer, pp. 195-222, 2016.

[4] P. Korshunov and S. Marcel, “DeepFakes: A New Threat to Face Recognition? Assessment and Detection”, arXiv:1812.08685, pp. 1-5, 2018.

[5] A. Dantcheva, C. Chen, A. Ross, “Can facial cosmetics affect the matching accuracy of face recognition systems?”, International Conference on Biometrics: Theory, Applications and Systems, pp. 391-398, 2012.

[6] Z. Akhtar and G. L. Foresti, “Face spoof attack recognition using discriminative image patches”, Journal of Electrical and Computer Engineering, pp. 1-14, 2016.

[7] M. Ferrara, A. Franco, D. Maltoni, “The magic passport”, IEEE International Joint Conference on Biometrics, pp. 1-7, 2014.

[8] J. Persch, F. Pierre, G. Steidl, “Examplar-Based Face Colorization Using Image Morphing”, Journal of Imaging, vol. 3, no. 4, pp. 1-13, 2017.

[9] I. Korshunova, W. Shi, J. Dambre, L. Theis, “Fast Face-swap Using Convolutional Neural Networks”, International Conference on Computer Vision, pp. 3697-3705, 2017.

[10] C. Seibold, W. Samek, A. Hilsmann, P. Eisert, “Detection of Face Morphing Attacks by Deep Learning”, Digital Forensic and Watermarking, pp. 1-13, 2017.

[11] T. Ahonen, A. Hadid, M. Pietikainen, “Face description with local binary patterns: application to face recognition”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, 2006.

[12] S. R. Dubey, “Face retrieval using frequency decoded local descriptor”, Multimedia Tools and Applications, pp. 1-21, 2017.

[13] R. Lan, Y. Zhou, Y. Y. Tang, “Quaternionic Local Ranking Binary Pattern: A Local Descriptor of Color Images”, IEEE Transactions on Image Processing, vol. 25, no. 2, pp. 566-579, 2016.

[14] L. Zhang, Z. Zhou, H. Li, “Binary Gabor pattern: An efficient and robust descriptor for texture classification”, IEEE international conference on image processing, pp. 81-84, 2012.

[15] V. Ojansivu, J. Heikkilä, “Blur insensitive texture classification using local phase quantization”, International conference on image and signal processing, pp. 236–243, 2008.

[16] J. Kannala and E. Rahtu, “BSIF: Binarized statistical image features”, International Conference on Pattern Recognition, pp. 1363-1366, 2012.

[17] J. Wu and J. M. Rehg, “CENTRIST: A visual descriptor for scene categorization”, IEEE transactions on pattern analysis and machine intelligence, no. 33, no. 8, pp. 1489-1501, 2011.

[18] A. Bosch, A. Zisserman, X. Munoz, “Representing shape with a spatial pyramid kernel”, ACM International Conference on Image and video retrieval, pp. 401-408, 2007.

[19] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[20] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, “Speeded-up robust features (SURF)”, Computer vision and image understanding, vol. 110, no. 3, pp. 346-359, 2008.

[21] P. Korshunov and S. Marcel, “Speaker inconsistency detection in tampered video”, European Signal Processing Conference, pp. 2375-2379, 2018.

[22] F. Yang, E. Shechtman, J. Wang, L. Bourdev, D. Metaxas, “Face Morphing using 3D-Aware Appearance Optimization”, Canadian Conference on Graphics Interface, p. 1-7, 2012.

[23] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, S.K. Nayar, “Face swapping: Automatically replacing faces in photographs”, ACM Trans. Graph., vol. 27, no. 3, p. 39:1-39:8, 2008.

[24] N.M. Arar, N.K. Bekmezci, F. Gney, H.K. Ekenel, “Real-time face swapping in video sequences: Magic mirror”, IEEE Signal Processing and Communications Applications Conference, p. 825-828, 2011.

[25] Z. Xingjie, J. Song, J. Park, “The image blending method for face swapping”, IEEE International Conference on Network Infrastructure and Digital Content, p. 95-98, 2014.

[26] T.M. den Uyl, H.E. Tasli, P. Ivan, M. Snijdewind, “Who do you want to be? real-time face swap”, IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), p. 1-1, 2015.

[27] S. Mahajan, L. Chen, T. Tsai, “Swapitup: A face swap application for privacy protection”, IEEE 31st International Conference on Advanced Information Networking and Applications, p. 46–50, 2017.

[28] Y. Nirkin, I. Masi, A.T. Tuan, T. Hassner, G. Medioni, “On face segmentation, face swapping, and face perception”, IEEE International Conference on Automatic Face Gesture Recognition, p. 98-105, 2018.

[29] R. Natsume, T. Yatagawa, S. Morishima, “FSNet: An Identity-Aware Generative Model for Image-based Face Swapping”, arXiv:1811.12666, p. 1-20, 2018.

[30] S. Suwajanakorn, S.M. Seitz, I. Kemelmacher-Shlizerman, “Synthesizing obama: Learning lip sync from audio”, ACM Trans. Graph., vol. 36, no. 4, p. 95:1–95:13, 2017.

[31] Y. Zhang, L. Zheng, V.L.L. Thing, “Automated face swapping and its detection”, IEEE 2nd International Conference on Signal and Image Processing (ICSIP), p. 15-19, 2017.

[32] A. Agarwal, R. Singh, M. Vatsa, A. Noore, “Swapped! digital face presentation attack detection via weighted local magnitude pattern”, IEEE International Joint Conference on Biometrics (IJCB), p. 659–665, 2017.

[33] D. Afchar, V. Nozick, J. Yamagishi, I. Echizen, “MesoNet: A Compact Facial Video Forgery Detection Network”, arXiv:1809.00888, p.1-7, 2018.

[34] Y. Li, M.C. Chang and S. Lyu, “In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking”, arXiv:1806.02877, p. 1-7, 2018.

[35] Z. Akhtar, D. Dasgupta, B. Banerjee, “Face Authenticity: An Overview of Face Manipulation Generation, Detection and Recognition”, International Conference on Communication and Information Processing (ICCIP), pp. 1-8, 2019.

978-1-7281-5092-5/19/$31.00 ©2019 IEEE