Faceswap Deepfakes Detection using Novel Multi-directional Hexadecimal Feature Descriptor

Qurat-ul-ain
Dept. of Computer Science
University of Engineering and Technology, Taxila
quratul.ain2@students.uettaxila.edu.pk

Ali Javed
Dept. of Software Engineering
University of Engineering and Technology, Taxila
ali.javed@uettaxila.edu.pk

Khalid Mahmood Malik
Dept. of Computer Science and Engineering,
Oakland University
mahmood@oakland.edu

2022 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST) | 978-1-6654-6051-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IBCAST54850.2022.9990223

***Abstract*— With the growing number of sophisticated deep learning algorithms and fake video generation applications, it is now possible to create highly realistic deepfake videos. Faceswap is the most commonly employed deepfakes approach, which is challenging to detect due to variations in the facial skin tone, illumination conditions, presence of accessories like glasses on the face, compression artifacts, etc. Existing local texture descriptors have achieved better performance on face recognition applications; however, they compute only the limited directional information while ignoring the magnitude details. This motivated us to develop a robust local texture descriptor to extract more directional and magnitude details from the adjacent pixels to effectively represent the video frames. For this purpose, we proposed a robust multi-directional hexadecimal feature descriptor (MDHFD) by combining the local hexadecimal pattern (LHeXDP) and Local Adjacent Neighborhood Magnitude Pattern (LANMP). LHeXDP calculates the orientation-based pattern by computing 1st- and 2nd-order derivatives at 0°, 45°, 90°, and 135° angles from each center pixel. LANMP computes the magnitude information from each central pixel to its adjacent pixels in horizontal, vertical, diagonal, and diagonal-back directions. Histograms of both the LHeXDP and LANMP are fused to compute a multi-directional feature vector, which is used to train a support vector machine to classify between the original or faceswap deepfakes video. We measured the performance of our system on the challenging faceswap subset of a diverse and large-scale Face Forensic++ and World Leaders datasets. Experimental results illustrate that the proposed method outperforms state-of-art methods for the detection of faceswap deepfakes.**

Keywords*— *Deepfakes, faceswap, LHeXDP, LANMP, magnitude-based patterns, orientation-based patterns, SVM.

I. INTRODUCTION

In recent years, artificial intelligence-based tools and applications have made it easier to create convincing synthetic multimedia content like speech, images, and videos. Generative adversarial networks (GANs) [3] and convolutional autoencoders [4] enable more accurate facial landmark identification, segmentation, and posture estimation. All these sophisticated deep learning techniques are now leading to generate highly realistic synthetic videos known as deepfakes. These deepfakes videos can be entertaining but can also use as an act of revenge and disinformation. The emergence of deepfakes in recent years has created much sensation globally due to its usage in spreading disinformation, disrupting government functioning, financial frauds, a threat to democracy, national security, etc. Among all types of deepfakes, faceswap is widely used to create synthetic videos that include replacing the face of a source person with that of the target person. Face swapping applications such as FakeApp [1] and Faceswap [2], have made it easier and faster to create deepfakes with more convincing results. Faceswap-oriented deepfakes are created to defame the reputation of famous people by portraying forged images and videos [5] or spreading disinformation for political revenge. Existing literature on faceswap-based deepfakes detection has explored various state-of-the-art hand-crafted descriptors-based approaches and end-to-end deep learning approaches.

Existing techniques [6-15] have used different handcrafted and local feature descriptors for faceswap detection. In [6], the faceswap detection method based on texture orientation and CNN was presented to increase the uniformity, but the performance of this system degrades when the diversity of the images increases. In [7], the SURF descriptor was used with the support vector machine (SVM) classifier for faceswap detection, but this approach is unable to recognize the manipulated videos. In [8], weighted local magnitude patterns were used with the SVM to detect the face swap deepfakes. A faceswapped dataset was also proposed for performance evaluation in [8], this method captures the details of spatial information but unable to capture the temporal features. In [9], 3D heads positions inferred from 2D facial landmarks were used to train the SVM classifier for recognition. This method [9] is unable to perform well on the blurred frames. Multimedia stream descriptors with the SVM were proposed in [10, 11] for the detection of synthetic faces, but this method was unable to detect the re-encoded videos. In [12], landmark features were used to train the SVM for deepfake detection and evaluated on a custom dataset of five US politicians. A deep vision algorithm was proposed in [13] to detect GAN-generated videos based on the duration of the intervened eye-blinking. In [14], the EAR method was used for facial landmarks identification of real-time eye blinking, but this system is unable to predict the frames having frequent eye blinking. The method in [13] provides better performance over [14], however unable to perform well on the videos that contain faces having irregular eye blinking patterns.

Existing approaches [15-22] have employed various DL-based techniques for faceswap deepfakes detection. In [15], a machine learning-based algorithm was proposed for face swap detection but failed to detect closed eye frames of FF++ videos. In [16], different pre-trained deep learning models were used i.e., ResNet50, Inception-V3, VGG-19, and VGG-16 for real and fake face detection datasets using Error level analysis features. It is to be noted that the performance of VGG-16 is better than all other pre-trained models for forgery recognition. In [17], different local descriptors i.e., LBP, LPQ, SIFT, and SURF were used for deepfakes detection. In

This work was supported by grant of Punjab HEC of Pakistan via Award No. (PHEC/ARA/PIRCA/20527/21).

978-1-6654-6051-4/22/$31.00 ©2022 IEEE

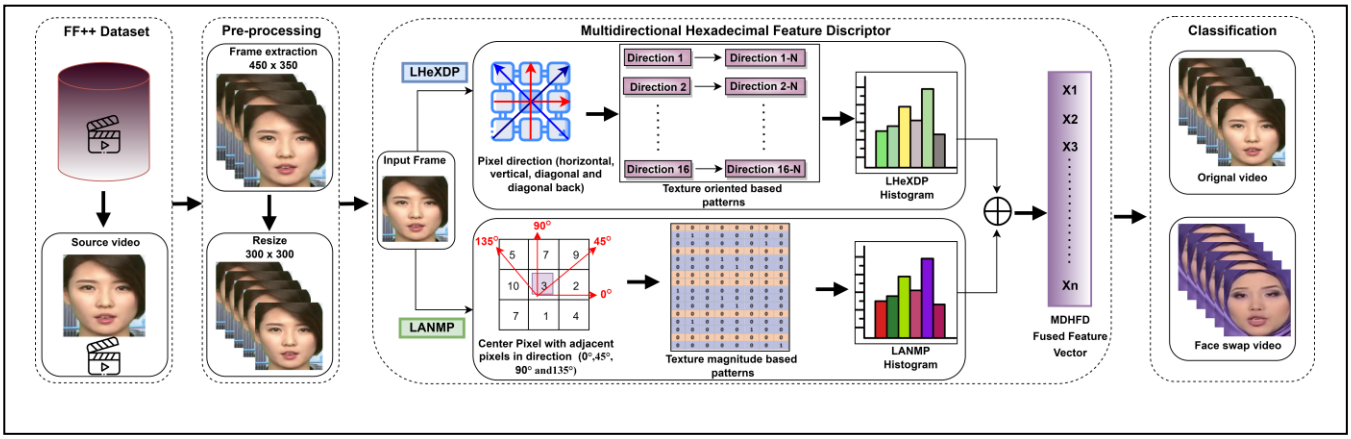

Fig.1. Architecture of Proposed Method.

[18], an end-to-end CNN model was employed for face frame detection using the bounded box regression. Performance of this method [18] degrades on the frames having non-frontal facial exposure. Vulnerabilities of VGG-16 and Face Net-based faceswap recognition systems were investigated in [19] and the results concluded that GAN-generated images are more difficult to recognize. In [20], deep features were used for deepfakes detection, however, unable to perform well on low-quality videos. In [21], CNN-based deep features were used for faceswap detection, but detection accuracy degrades on the unseen data. In [22], frame-level features were extracted using CNN and used to train the SVM for deepfakes detection. This method [22] was unable to perform better on the compressed videos. In [23], a comprehensive review of all major types of deepfakes, specially faceswap generation and detection is presented in detail.

Literature shows that existing techniques, especially local descriptors [17] are unable to detect face swapped deepfakes on unseen data [7] and GAN-generated videos [12]. Methods based on local texture descriptors [17] like LBP, SURF, and LTP compute only limited directional information and disregard the magnitude information. The performance of these descriptors can be further improved by gathering more directional and higher-magnitude information. Our motivation was to develop a robust local texture descriptor by capturing more directional and higher-magnitude details from the adjacent pixels to effectively represent the frames of real and faceswap videos. The following are the significant contributions of this work:

1. We propose a novel multi-directional hexadecimal feature descriptor (MDHFD) to effectively capture the texture orientation and magnitude information from the video frames.
2. We propose an effective faceswap deepfakes detection system that is robust on videos containing variations of the facial skin tone of people having different races, illumination conditions, presence of accessories like glasses on the face, and loss of details due to compressed video resolution.
3. Rigorous experimentation was performed on two diverse deepfakes datasets including the cross-corpora evaluation to test the generalizability of our method.

II. PROPOSED METHOD

This section provides a discussion on the proposed method for faceswap deepfakes detection. The architecture of the proposed framework is shown in Fig. 1.

A. Face detection

We used Multi-task Cascaded Convolutional Networks (MTCNN) [18] for frame-level extraction. MTCNN is designed to detect and extract only the face portion from the input video. The method detects the facial landmark locations, coarse to fine details such as the eyes, nose, and mouth. In comparison with other face detectors like Haar Cascade, MTCNN detects the faces precisely in the presence of occlusion and varying illumination conditions.

B. Feature extraction

After face detection from the input video frames, we obtain the frames containing the frontal faces exposure. Next, we need to better capture the traits of these facial images in real and fake videos. For this purpose, we propose a novel local texture descriptor MDHFD to extract the multi-directional texture features from each frame. The details of the proposed descriptor are given in the subsequent section.

1) Multi-directional hexadecimal feature descriptor

To better capture the attributes of real and fake images from the extracted facial frames, we propose a novel feature extractor, a multi-directional hexadecimal feature descriptor for frame representation. The proposed descriptor is comprised of the direction and magnitude-based features. The orientation patterns effectively extract discriminative information from the frames and compute directional information by taking derivative from the center pixel to its neighbouring horizontal, diagonal, diagonal back, and vertical derivatives. Furthermore, we compute the patterns based on magnitude which captures additional useful information. Each resultant feature vector is a combination of magnitude and orientation patterns.

To calculate the orientation, we employed a local hexadecimal feature descriptor (LHeXDP). For a given frame $F(x, y)$, we compute the 1st order derivative at the grayscale value of the surrounding pixels along with directions, as shown in (1). The frame is translated into 16 different values from which the directions are determined. We construct

texture orientation-based pattern for each pixel by comparing the 1st order derivatives of the center pixel direction with the directions of all the eight surrounding neighbors. After that, the orientation pattern is subdivided into 16 binary patterns by division of 2nd order derivatives. Similarly, for the nth directional pattern, the derivatives of surrounding directions are computed by taking the difference with the center pixel.

$$F_{\alpha}^{1}\,(d_{h,d,v,db})|\alpha = 0^0, 45^0, 90^0, 135^0 \quad (1)$$

In (1), $(d_{h,d,v,db})$ represents the direction from the center pixel to adjacent pixels that are horizontal, diagonal, vertical, and diagonal back. Total 16 values are computed w.r.t the center pixel. $F_{dir}^{1}\,(d_s)|_{s=1\ldots,8}$ represents 8 surrounding pixels with respect to 3×3 adjacent pixels.

Taking the 2nd order derivative of the central pixel, we obtained 8-bit directions with all the eight neighbouring directions (s=1-8) using the (2) and (3). Furthermore, the directional pattern is separated into 15 binary patterns. The central pixel with its corresponding pixels is computed using the (1). The 2nd order is distributed into 16 binary patterns computed as shown in (4). In this way, we calculate the directional patterns.

$$LHXDP^2 = \begin{Bmatrix} T_1\left(F_{dir}^{1}.(d_c).F_{dir}^{1}.(d_1)\right).T_1\left(F_{dir}^{1}.(d_c).F_{dir}^{1}.(d_2)\right)\ldots \\ \ldots,.\ T_1\left(F_{dir}^{1}.(d_c).F_{dir}^{1}.(d_s)\right)|_{s=8} \end{Bmatrix} \quad (2)$$

$$F_{dir}^{1}(d_c) \times F_{dir}^{1}(d_s) = \begin{cases} 0, & F_{dir}^{1}.(d_c) = F_{dir}^{1}.(d_s) \\ F_{dir}^{1}.(d_s), & otherwise \end{cases} \quad (3)$$

$$LHXDP^2|_{directions=1,2,3,4,5,6,7,8,9,10,11,12,13,15,16} = \sum_{s=0}^{s} 2^{(s-1)}\, x\, T_1(LHXDP^2(d_c))|_{directions=\propto} \quad (4)$$

For magnitude information, we compute the magnitude-based Local Adjacent Neighborhood Magnitude Pattern (LANMP) using the horizontal, diagonal, and vertical by taking 1st order derivatives that captures edge information of each frame in detailed.

$$M_1^1.(d_s) = \sqrt{\left(F_{0^0}^{1}(d_s)\right)^2 + \left(F_{45^0}^{1}(d_s)\right)^2 + \left(F_{105^0}^{1}(d_s)\right)^2 + \left(F_{135^0}^{1}(d_s)\right)^2} \quad (5)$$

$$LANMP^2 = \sum_{s=1}^{s} 2^{(s-1)} \times R\,(M_1^1.(d_s) - M_1^1.(d_c))|_{s=8} \quad (6)$$

$$R(x) = \begin{cases} 1, & x >= 0 \\ 0, & otherwise \end{cases} \quad (7)$$

In (7), R is the function that discriminates between the magnitude of surrounding neighbors' pixels and the center pixel based on parameter x. The magnitude descriptor assigns a value of 1 if the magnitude of the surrounding pixel is higher than the magnitude of central pixel. After extracting the local patterns (LHXDP and LANMP) of each frame, we obtain the histogram of both patterns. The proposed MDHDF is formed by fused histograms of LHXDP and LANMP.

$$\text{Hist}^{MDHFD} = [\text{Hist}^{LHXDP}||\text{Hist}^{LANMP}] \quad (8)$$

C. Classification

We employed SVM with the proposed descriptor for the classification of the original and face swapped videos. The reason we adopted the SVM is that it has the optimization property of achieving a global minimum and has a property to map non-linearly separable data into higher space. We used Radial Basis Function (RBF) kernel approach as the data represented by the feature descriptor was not linearly separable. This kernel is suitable for the linear separation of higher dimensions. SVM adopted the labelled frames from the training set and performance for the classifier was evaluated on the unseen data. We also used other classifiers like Bagged Trees Ensemble, Fine Tree, and Narrow Neural Network, but SVM achieved better results comparatively. For experimentation using the SVM, we set the kernel function to gaussian, box constraint level to 1, one vs one multiclass method, and kernel scale to 4.5, as we attained the best results on these settings.

III. Experiment Setup

This section presents the details of dataset and different experiments conducted to measure the performance of our method.

A. Dataset

We evaluated the performance of the proposed method on the original and face swapped subset of a challenging Face Forensics++ dataset (FF++) [22] and World Leaders dataset (WLDR) [12]. FF++ subsets comprised of 1000 original and faceswap videos sequences. The data was gathered from 977 YouTube videos. Each video in a subset contains one person having a non-occlusion frontal face but it is challenging due to variations in the face skin tone of people having different races, illumination conditions, presence of accessories like glasses on the face, loss of details due to compressed video resolution. WLDR dataset [12] is comprised of YouTube videos of world-famous leaders (Obama, Clinton, Trump, etc.) having their original, comedic impersonators, faceswap, and lip-sync subsets. We evaluated our proposed model on the faceswap subset of the WLDR dataset. Leaders are talking throughout the video segments; each video contained only one face, and the camera was stationary with slight variations in zooming. We split both datasets into 80:20, where 80% of the frames are used for training and rest 20% for the testing.

B. Performance Evaluation of proposed method

To evaluate the effectiveness of our method for faceswap deepfakes detection, we designed an experiment to evaluate the performance of our method on the original and faceswap subset on both the FF++ and the WLDR datasets. We resized all frames to 300 × 300 and extracted the features using our MDHFD for both the training and testing set. These features are then used to train the SVM classifier for video deepfakes detection. SVM produce a better result on proposed MDHFD features using a gaussian kernel. More specifically, we achieved an accuracy of 92.3% and AUC of 0.99 on the FF++ dataset, whereas achieved an accuracy of 96.9% and AUC of 1.00 on the WLDR dataset.

C. Confusion Matrix Analysis

To better investigate the false-acceptance and rejection scenarios, we designed a confusion matrix analysis to depict the classification performance of our method on both datasets as shown in Figs. 2 and 3. From the classification details presented in Fig. 2, we can see that our method incorrectly

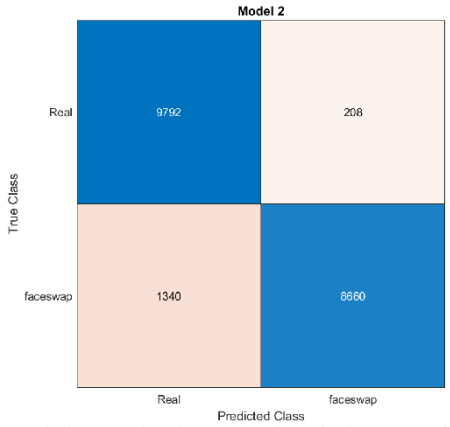

Fig. 2. Confusion Matrix of a Proposed Method on FF++ dataset.

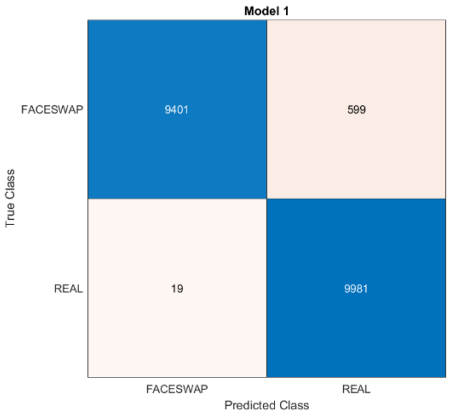

Fig.3. Confusion Matrix of a Proposed Method on WLDR dataset.

classifies only 208 samples as fake and 1340 samples as real on the FF++ dataset. The 12% false-negative rate signifies a remarkable recall rate of 87.9% achieved by the proposed method. Similarly, 2% of false positives show a better precision rate of 97.9%. For the WLDR dataset, our method incorrectly classifies only 19 samples as fake and 599 samples as real. Moreover, our method achieved a recall rate of 94.3% with a 0.05% false-negative rate. Similarly, 0.02% false positives show the best precision rate of 99.8% of our method for faceswap deepfakes detection.

D. ROC Curve Analysis

In order to determine the capability of our proposed model to differentiate between the original and faceswap samples we created the receiver operating characteristics (ROC) curve. From the ROC curves in Figs. 4. and 5, it can be observed that due to the directional and magnitude nature of the proposed descriptor with the SVM classifier, it attains higher results. From Table I, the AUC of the proposed method on both the FF++ and the WLDR datasets using different classifiers can be observed. Using SVM with proposed features we achieved the best AUC of 0.99 on the FF++ dataset. Whereas the Bagged Trees Ensemble, Narrow Neural Network, KNN, and Decision Tree achieved the AUC of 0.97, 0.94, 0.92, and 0.82, respectively. On the WLDR dataset, we achieved the best AUC of 1.00 with SVM, whereas the Bagged Trees Ensemble, Narrow Neural Network, KNN, and Decision Trees achieved AUC of 1.00, 1.00, 0.99, and 0.98, respectively.

E. Performance comparison with other classifiers

To evaluate the performance of SVM against other classifiers for faceswap deepfakes detection, we designed an experiment to test the performance of our MDHFD descriptor on various conventional machine learning and deep learning classifiers. More precisely, in Table I, we compared the performance of SVM with these classifiers i.e., Bagged Trees Ensemble, Narrow Neural Network, KNN, and Decision Trees. We employed our MDHFD features to train each classifier separately. From this experiment, we observed that on FF++, SVM performed the best in comparison with other classifiers by achieving an accuracy of 92.3%. Whereas the Bagged Trees Ensemble, Narrow Neural Network, and KNN achieved an accuracy of 89.9%, 85.49%, and 80.4%, respectively. The decision trees tune at a fine level attained the lowest accuracy of 77.3%. Similarly, on the WLDR dataset, SVM achieved the highest accuracy of 96.9%, whereas, the Bagged Trees Ensemble, Narrow Neural Network, KNN, and Decision Trees achieved an accuracy of 95.6%, 95.5%, 94.5%, and 91.9%, respectively. The decision trees with both datasets especially on FF++ produce the lowest accuracy than other classifiers because decision trees are more prone to overfitting. The overall accuracies of both datasets using different classifiers can be seen in Table I.

F. Cross-Corpora Evaluation

To test the generalizability of our method, we designed a two-stage cross-corpora evaluation experiment using the proposed features. In the first stage of this experiment, we

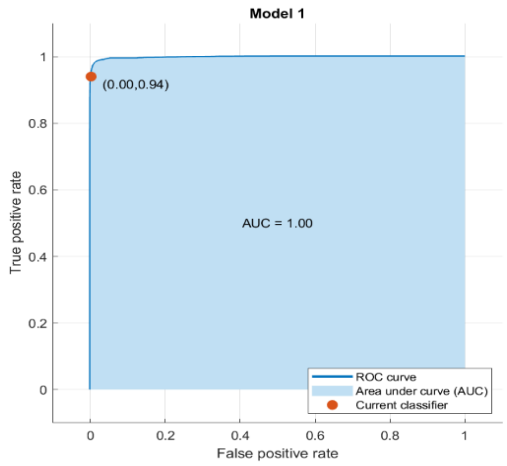

Fig. 4. ROC curve of Proposed Method on WLDR Dataset.

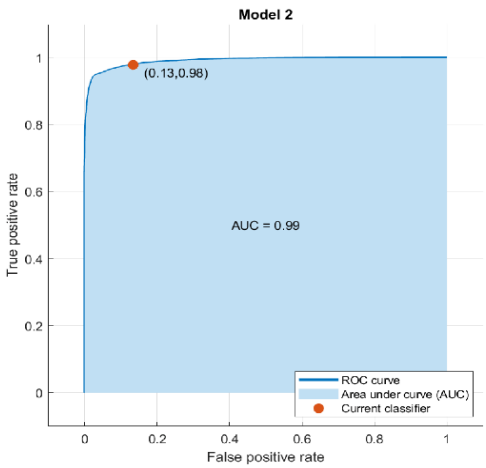

Fig. 5. ROC curve of Proposed Method on FF++ Dataset.

TABLE I. PERFORMANCE EVALUATION OF PROPOSED METHOD ON DATASETS USING DIFFERENT CLASSIFIERS.

Classifiers	FF++ Dataset		WLDR Dataset	
	Accuracy	AUC	Accuracy	AUC
SVM	92.3%	0.99	96.9%	1.00
Bagged Trees Ensemble	89.9%	0.97	95.6%	1.00
Narrow Neural Network	85.49%	0.94	95.5%	1.00
Fine Trees	77.3%.	0.82	91.9%	0.98
KNN	80.4%	0.92	94.5%	0.99

used WLDR dataset to train our model and evaluated it on the FF++ dataset. We achieved 98.8% training accuracy but attain a less accurate result of 49.6% and an AUC of 0.69 on the test set. In the second stage of this experiment, we used the FF++ dataset for training and the WLDR dataset for testing and achieved an accuracy of 93.4% on training but attained less accurate results of 50% accuracy and 0.50 AUC on the test set. Cross-corpora experiments attained lower results because both FF++ and WLDR dataset are different from each other. For instance, videos are diverse in terms of illumination and occlusion conditions, presence of accessories like glasses on the face, and loss of details due to compressed video resolution.

G. Performance evaluation with Contemporary methods

To analyse the effectiveness of the proposed method for deepfakes detection, we compared it with the existing state-of-art methods [8,9,10,12,13,15,16,21,22], and the results are shown in Table II. AUC of [15] is 0.78, which is worst in comparison with other methods, while [10] performed the second-best having different AUC on different classifiers; however, the proposed method outperforms with 0.99 AUC on the FF++ dataset and 1.00 AUC on the WLDR dataset. Similarly, the accuracy of [21] is the worst, and [8] achieved the second-best accuracy, whereas the proposed method achieved the best accuracy of 92% on the FF++ dataset and 96.9% on the WLDR dataset, respectively. From this comparative analysis, it can be determined that our proposed method achieves superior detection performance over the contemporary methods. Therefore, we claim that the proposed method (MDHFD-SVM) can effectively be used to classify the faceswap deepfakes detection.

VI . CONCLUSION

In this research work, we have presented a novel texture feature descriptor to reliably capture the orientation and magnitude-oriented details from the input video frames that are then used to detect the faceswap deepfakes. The proposed method better addresses the problem of faceswap-based deepfakes detection under challenging conditions such as variations in the face skin tone of people having different races, illumination conditions, presence of accessories like glasses on the face, etc. We evaluated the performance of the proposed method on the faceswap subset of Face Forensics++ and the WLDR dataset that contains all the aforementioned challenges. Experimental results on both datasets illustrate the effectiveness of the proposed method over the state-of-the-art methods for better detection of faceswap deepfakes. In the future, we plan to extend our work to detect multiple types of deepfakes and will try to improve our results on cross-corpora evaluation.

TABLE II. PERFORMANCE EVALUATION OF PROPOSED METHOD WITH STATE-OF-THE-ART METHODS.

	Methods	Technique	Results
AUC	**Proposed Method**	**MDHFD + SVM (FF++)** **MDHFD + SVM (WLDR)**	**0.99** **1.00**
	[9]	SVM Classifier+ landmarks	0.89
	[10]	SVM+ Multimedia stream descriptor	0.93
	[12]	SVM classifier	0.96
	[15]	texture energy-based features + (MLP, LogReg)	0.851 0.784
Accuracy	**Proposed Method**	**MDHFD + SVM (FF++)** **MDHFD + SVM (WLDR)**	**92.3%** **96.9%**
	[16]	ELA+(VGG16, ResNet50, InceptionV3, VGG19)	64.49% 53.64% 57.25% 60.63%
	[8]	SURF + SVM	92%
	[13]	EAR+ landmarks	87.5%
	[21]	Deep Features + CNN	83.71%
	[22]	Deep Features + SVM	90.29%

ACKNOWLEDGMENT

This work was supported by the Punjab Higher Education Commission of Pakistan via Award No. PHEC/ARA/PIRCA/20527/21. We would like to thank Prof. Hany Farid from University of California Berkeley to provide us their World Leaders Dataset for performance evaluation.

REFERENCES

[1] FakeApp 2.2.0, Available at: https://www.malavida.com/en/soft/fakeapp/. Accessed: September 18, 2020.

[2] Faceswap: Deepfakes software for all, Available at: https://github.com/deepfakes/faceswap. Accessed: September 08, 2020.

[3] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional generative adversarial networks. arXiv:1702.01983, Feb. 2017.

[4] A. Tewari et al. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 1274–1283, Oct. 2017. Venice, Italy.

[5] J. F. Boylan, "Will deep-fake technology destroy democracy?" The New York Times, Oct, vol. 17, 2018

[6] A. Khodabakhsh, R. Ramachandra, K. Raja, P. Wasnik, C. Busch, in 2018 International Conference of the Biometrics Special Interest Group (BIOSIG). Fake face detection methods: can they be generalized? (IEEE, 2018). https://doi.org/10.23919/biosig.2018.8553251.

[7] Y. Zhang, L. Zheng, and V. L. Thing, "Automated face swapping and its detection," in 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), 2017, pp. 15-19: IEEE.

[8] A. Agarwal, R. Singh, M. Vatsa, A. Noore, in 2017 IEEE International Joint Conference on Biometrics (IJCB). Swapped! Digital face presentation attack detection via weighted local magnitude pattern (IEEE, 2017).

[9] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in ICASSP 2019- 2019 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2019, pp. 8261-8265: IEEE.

[10] D. Güera, S. Baireddy, P. Bestagini, S. Tubaro, and E. J. Delp, "We Need No Pixels: Video Manipulation Detection Using Stream Descriptors," arXiv preprint arXiv:1906.08743, 2019.

[11] K. Jack, "Chapter 13-MPEG-2," Video Demystified: A Handbook for the Digital Engineer, pp. 577- 737.

[12] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 38–45, 2019.

[13] T. Jung, S. Kim, and K. Kim, "DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern," IEEE Access, vol. 8, pp. 83144-83154, 2020.

[14] T. Soukupova and J. Cech, "Eye blink detection using facial landmarks," in 21st computer vision winter workshop, Rimske Toplice, Slovenia, 2016.

[15] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), 2019, pp. 83-92: IEEE.

[16] Qurat-ul-ain, N. Nida, A. Irtaza, and N. Ilyas, "Forged Face Detection using ELA and Deep Learning Techniques," *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, 2021, pp. 271-275, doi: 10.1109/IBCAST51254.2021.9393234.

[17] Akhtar, Z., & Dasgupta, D. (2019, November). A comparative evaluation of local feature descriptors for deepfakes detection. In *2019 IEEE International Symposium on Technologies for Homeland Security (HST)* (pp. 1-5). IEEE.

[18] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499-1503, 2016.

[19] P. Korshunov, S. Marcel, Deepfakes: a new threat to face recognition? assessment and detection. arXiv preprint arXiv:1812.08685 (2018).

[20] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: a compact facial video forgery detection network," in 2018 IEEE International Workshop on Information Forensics and Security (WIFS), 2018, pp. 1-7: IEEE.

[21] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," in 2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS), 2019, pp. 1-8.

[22] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "Faceforensics++: Learning to detect manipulated facial images," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1-11.

[23] Masood, M., Nawaz, M., Malik, K. M., Javed, A., Irtaza, A., & Malik, H. (2022). Deepfakes Generation and Detection: State-of-the-art, open challenges, countermeasures, and way forward. *Applied Intelligence*, 1-53.