Deepfake Detection with Clustering-based Embedding Regularization

Kui Zhu
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
zhukui@bupt.edu.cn

Bin Wu*
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
wubin@bupt.edu.cn

Bai Wang
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
wangbai@bupt.edu.cn

***Abstract*—In recent months, AI-synthesized face swapping videos referred to as deepfake have become an emerging problem. False video is becoming more and more difficult to distinguish, which brings a series of challenges to social security. Some scholars are devoted to studying how to improve the detection accuracy of deepfake video. At the same time, in order to conduct better research, some datasets for deepfake detection are made. Companies such as Google and Facebook have also spent huge sums of money to produce datasets for deepfake video detection, as well as holding deepfake detection competitions. The continuous advancement of video tampering technology and the improvement of video quality have also brought great challenges to deepfake detection. Some scholars have achieved certain results on existing datasets, while the results on some high-quality datasets are not as good as expected. In this paper, we propose new method with clustering-based embedding regularization for deepfake detection. We use open source algorithms to generate videos which can simulate distinctive artifacts in the deepfake videos. To improve the local smoothness of the representation space, we integrate a clustering-based embedding regularization term into the classification objective, so that the obtained model learns to resist adversarial examples. We evaluate our method on three latest deepfake datasets. Experimental results demonstrate the effectiveness of our method.**

***Index Terms*—face swapping, deepfake detection, clustering-based, regularization**

I. Introduction

At present, intelligent video recognition based on deep learning has been applied to various fields of the country and society, such as network content supervision, intelligent video surveillance and autonomous vehicles. People post photos and videos every day on popular social software and websites such Weibo, Facebook, Instagram and Twitter. However, manipulation of visual content has now become ubiquitous. In the past two years, AI-algorithms for face swap and many other video manipulation methods have been proposed. Many such tools are publicly available as open source software, such as DeepFake [1], Face2Face [2], FaceSwap [3], FakeApp [4]. In order to make the face-swapping more realistic, scholars have done a lot of research, and more and more deep learning techniques are applied to this application. For example, some work based on GAN has achieved very good results [5]–[7]. Li et al. [8]propose a novel two-stage framework, called FaceShifter, for high fidelity and occlusion aware face swapping. Neekhara et al. demonstrate that it is possible to bypass DNNs detectors by adversarially modifying fake videos synthesized using existing Deepfake generation methods [9]. Gandhi et al. [10] uses adversarial perturbations to enhance deepfake images and fool common deepfake detectors. Carlini et al. [11] develop five attack case studies on a state-of-the-art classifier that achieves an area under the ROC curve (AUC) of 0.95 on almost all existing image generators, when only trained on one generator. The continuous advancement of video tampering technology and the improvement of video quality have also brought great challenges to deepfake detection.

Detection of manipulated visual content becomes a research hotspot at the same time. Google's researchers published the AI Principles and announced that they are committed to developing AI best practices to mitigate the potential for harm and abuse. They announced a dataset of synthetic speech in support of an international challenge to develop high-performance fake audio detectors in 2018, and they announced the release of a large dataset of visual deepfakes last year [12]. AWS, Facebook, Microsoft, the Partnership on AI's Media Integrity Steering Committee, and academics have come together to build the Deepfake Detection Challenge (DFDC). The goal of the challenge is to spur researchers around the world to build innovative new technologies that can help detect deepfakes and manipulated media [13]. Tolosana et al. [14]provides an exhaustive analysis of both 1st and 2nd DeepFake generations in terms of facial regions and fake detection performance. Korshunov et al. [15] do experiments and demonstrate that GAN-generated Deepfake videos are challenging for both face recognition systems and existing detection methods, and the further development of face swapping technology will make

*This author is a corresponding author of this research.

978-1-7281-9558-2/20/$31.00 ©2020 IEEE
DOI 10.1109/DSC50466.2020.00046

it even more so.

Deepfake technology also plays a unique role in some specific fields. For example, Zhu et al. [16] use deepfake technology for face swapping to protect privacy in medical videos. So we can't say that deepfake technology is harmless and unprofitable

In this work, we propose a new method with clustering-based embedding regularization for deepfake detection. We use open source algorithms to simulate the process of the process of generating artifacts during the deepfake generation process using simple image processing operations on a image, and make it as a generated example. We train the Xception network for classification, using positive samples, negative samples, and generated samples as input samples. Class number is set to 3 during training process, and in the testing process the generated samples are also classified as negative samples to improve the classification effect. At the same time, a regularization loss is added during the training process to ensure the inter-class distance and intra-class smoothness of the embedded space. Our contributions are as follows:

1. We simulate the process of the deepfake videos' generation process using simple image processing operations on a image, and we use the simulated samples as a additional class to train the Xception networks. When testing, the simulated samples are classified as negative samples.
2. We introduce a clustering-based embedding regularization into adversarial learning, which further guarantees the inter-class distance and the intra-class smoothness in the embedding space, therefore improves the robustness of our model.
3. Experimental results on UADFV, Celeb-DF and DeepFakeDetection datasets demonstrate the effectiveness of our methods in deepfake detection.

II. RELATED WORK

With the development of face swap and other video tampering technologies, the detection of deepfake has gradually become a research hotspot. Some researchers use the data collected on social software and the Internet to conduct research, while some researchers have released datasets specifically for deepfake detection, such as UADFV [17], Celeb-DF [18], DeepFakeDetection, DeepFake-TIMIT [19] and FaceForensics++ [20].

Some researchers determine whether a video is a deepfake video by detecting specific artifacts commonly in the deepfake video. For example, Xin et al. [17] compare head poses estimated using all facial landmarks and those estimated using only the central region to expose AI-generated fake face images or videos. Li et al. [21] also attempt to detect eye blinking in the videos, which is a physiological signal that is not well presented in the synthesized fake videos. Falko et al. [22] propose a set of straightforward features in eyes, teeth and facial contours for detecting generated faces, deepfakes, and Face2Face images. Guarnera et al. [23] focus on the analysis of Deepfakes of human faces with the objective of creating a new detection method able to detect a forensics trace hidden in images: a sort of fingerprint left in the image generation process.

In addition, some other researchers use Convolutional Neural Networks(CNN) or a combination of CNN and Recurrent Neural Network(RNN) to detect deppfake videos. Andreas et al. [13] propose the dataset of facial forgeries, FaceForensics++, and detect deepfake videos by training the Xception model on FaceForensics++ dataset. David et al. [24] propose a temporal-aware pipeline to automatically detect deepfake videos. Their system uses a CNN to extract frame-level features which are then used to train a RNN that learns to classify if a video has been subject to manipulation or not. Ekraam et al. [25] propose to leverage temporal artefacts as a means for indication of abnormal faces in a video system. Li et al. [26] propose a novel PatchPair Convolutional Neural Networks (PPCNN) to distinguish Deepfake videos or images from real ones. Ruiz et al. do adversarial training for generative adversarial networks (GANs) as a first step towards robust image translation networks [27]. Mittal et al. [28] present a learning-based multimodal method for detecting real and deepfake videos. Montserrat et al. [29] introduce a method based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) that extracts visual and temporal features from faces present in videos to accurately detect manipulations. Vignesh K et al. [30] propose a framework which utilizes a convolutional neural system (CNN) to remove outline level highlights. These highlights are then used to prepare a repetitive neural net-work (RNN) that figures out how to characterize if a video has been sub-ject to control or not. Amerini et al. [31] propose a new forensic technique which can discern between fake and original video sequences. Unlike other methods which resorts at single video frames, they propose the adoption of optical flow fields to exploit possible inter-frame dissimilarities.

Many others use techniques from other fields to resolve deepfake videos. Two-stream DNN-based forgery detection method [32] which trains GoogleNet to detect tampering artifacts and trains a patch based triplet network to leverage features capturing local noise residuals. Darius et al. [33] proposes to adopt an intermediate approach using a deep neural network with a small number of layers and two networks called MesoNet. Huy et al. [34] design a convolutional neural network that uses the multi-task learning approach to simultaneously detect manipulated images and videos and locate the manipulated region for each query. Kumar et al. [35]analyze several deep learning approaches in the context of deepfakes classification in high compression scenario and demonstrate that a proposed approach based on metric learning can be very effective in performing such a classification. Vignesh K et al. [30] introduce the image classification models to apprehend the features from each deepfake video frames. Sohrawardi et al. [36] propose a system that will robustly and efficiently enable users to determine whether or not a video posted online is a deepfake. They approach the problem from the journalists' perspective and work towards developing a tool

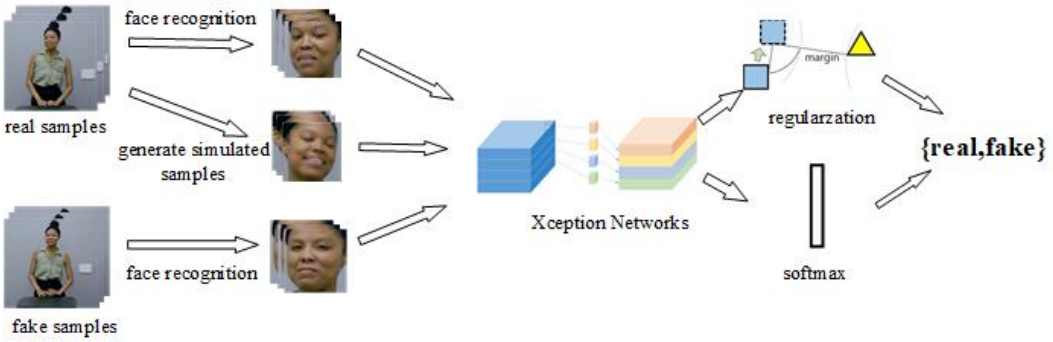

Fig. 1. Overview of our deepfake detection model. For real samples and fake samples, we extra face region using face recognition algorithm and generate simulated samples from real samples. Add regularization loss to the Xception network and finally get a binary classifier.

TABLE I
THE DESCRIPTION OF THE DATASETS, INCLUDING THE DIVISION OF TRAINING AND TEST SETS, THE NUMBERS OF VIDEOS AND FRAMES.

	train		test	
	real	fake	real	fake
UADFV	35 videos (13976 frames)	35 videos (13638 frames)	14 videos (3353 frames)	14 videos (3353 frames)
Celeb-DF	370 videos (158992 frames)	733 videos (290043 frames)	38 videos (16409 frames)	62 videos (22834 frames)
DeepFakeDetection	254 videos (202723 frames)	2148 videos (1678558 frames)	109 videos (94437 frames)	920 videos (681550 frames)

to fit seamlessly into their workflow.

Li et al. [37] detect deepfake videos by exposing the artifacts caused by the post-processing operations in the deepfake video generation. They use simple image processing operations on a image to make it as negative example. They are the most related work to ours. However, our method mainly differs from them in two aspects: (1) Their method focuses on detecting artifacts that are common in deepfake videos without using label information for negative samples. We use positive samples, negative samples, and generated samples as training inputs, and increase the use of negative sample label information. (2) We add a margin-based regularization loss during the training process to guarantee the inter-class distance and the intra-class smoothness in the embedding space,

III. PREPROCESSING AND NETWORKS

Li et al. [37] perform experiments on the UADDV and DeepfakeTIMIT datasets respectively, and the showe that ResNet networks have about 10% better performance compared to VGG16. Andreas et al. [20] experiment on FaceForensics++ datasets and achieve good results using Xception networks. We conducted a simple experimental comparison, and finally found that the Xception network can achieve better results. In addition, the generation process of the simulation samples and the regularization loss are described in detail below. We present our new method with clustering-based embedding regularization in this section, including simulated samples' generated process and the define of clustering-based embedding regularization. The model diagram is shown in Figure 1.

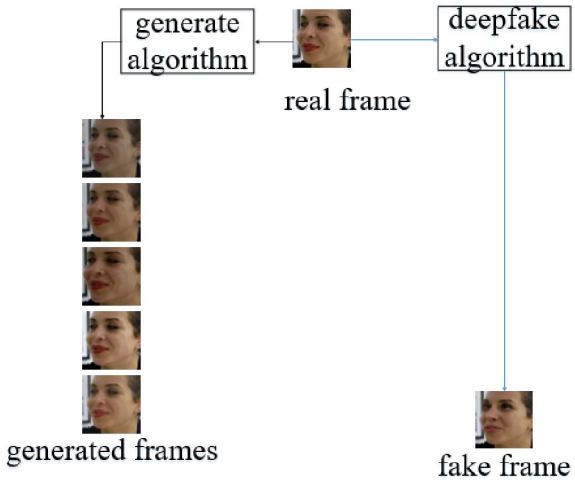

Fig. 2. The process of generating fake samples and simulated samples

A. Generate Simulated Samples

Li et al. [37] find that current deepfake algorithm can only generate images of limited resolutions, which need to be further warped to match the original faces in the source video. However, during the post-processing of the generation,

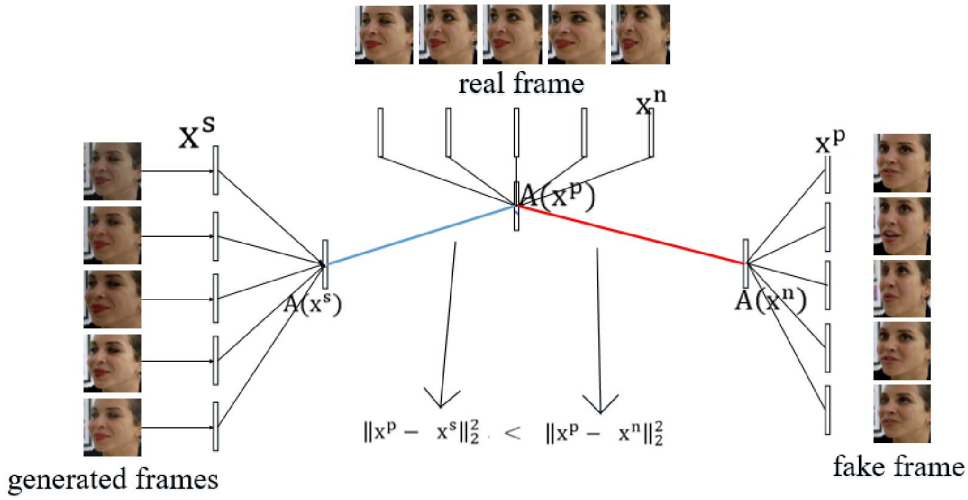

Fig. 3. The training process of a batch size sample

certain traces are often left in the generated video which called artifacts. They use real samples and simulated samples as the input to the network, and regard the deepfake detection task as a binary classification problem. Experiments prove that their method has achieved good experimental results in some low-quality tampering videos. However, with the development of video tampering technology and the improvement of video quality, the classification effect is often not very good. Their method only focused on the artifacts generated during the video processing, and ignored the label information of the dataset. With the development of deepfake detection research, many labeled data sets have appeared. The use of label information will help us get better performance on deepfake detection. In our method, we use their open source algorithm to generate simulation samples and use the generated samples as a new class of samples. In the training process, we input real samples, negative samples, and simulated samples into the neural network for training at the frame level. we add the output probabilities of the negative samples and the simulated samples as the output probabilities of the negative samples.

In order to reduce the training and testing time, we first extract the faces and surrounding areas in each frame of real videos and deepfake videos. The generation of simulated samples of a frame includes the process of face extraction, random size scaling, Gaussian blur, and rescaling to put back to the original picture. In order to increase the diversity of training samples, we set the color information of the sample to be randomly changed, including brightness, contrast, distortion and sharpness. Similarly we set to randomly extract the area around the face. For each real image, we repeat the simulation generation process 10 times, and make the generated pictures as simulated samples.

As shown in Figure 2, we use generation algorithm to get multiple simulated samples. For a frame in the real sample, we will get a series of simulated frames with smaller differences.

B. Clustering-based Embedding Regularization

Embedding regularization has achieved good effect in many fields. Zhou et al. [32] propose a two-stream network architecture to capture both tampering artifact evidence and local noise residual evidence. By training the triplet network, they ensure that a pair of patches from the same image are closer in the learned embedding space, while the distance between a pair of patches from two different images is large. However, the positive patch and the negative patch from one anchor may be extracted from two completely different pictures. And the positive patch is extracted from the face area in one picture, while the negative patch is extracted from the non-face area of another picture. Zhong et al. [38] think that the Deep neural networks(DNNs) are highly vulnerable to adversarial attacks. So that they propose to improve the local smoothness of the representation space, by integrating a margin-based triplet embedding regularization term into the classification objective, so that the obtained model learns to resist adversarial examples. On this basis, we improve the calculation of regularization in our work. For each training iteration, we choose the vector obtained by the last layer of the neural network as the embedding representation of the input samples. As shown in Figure 3, With a tuple of input samples x^p, x^n, x^s, including one real sample's embedding representation x^p, one fake sample's embedding representation x^n and one simulated sample's embedding representation x^s. Obviously, x^s is derived from x^p through some simple transformations, including artifacts generated during resize operation and some minor changes in brightness, contrast, distortion and sharpness. We define this transformation as $\triangle x$. So that we get the formula(1). The blue line in the figure represents the distance between real frame and simulated

frame, and the red line represents the distance between real samples and fake samples. We believe that in the generation process of simulated samples, our modification of real samples is less than that in the deepfake algorithm. So that We would like the following constraint to be satisfied:

$$x^p + \triangle x = x^s \tag{1}$$

$$\|x^p - x^n\|_2^2 < \|x^p - x^s\|_2^2 \tag{2}$$

But in one training iteration, the input of the neural network is often not only one tuple. During the training process, we input one batch size samples at a time. In one batch size, we set the ratio of real samples, fake samples and simulated samples to 1: 1: 1. Often we calculate the constraint of each tuple separately and then sum them, but this calculation method has great randomness. We introduce the idea of clustering in regularization's calculation. Suppose we have n positive samples, n negative samples, and n simulated samples during one training process. We calculate the average embedding representation of all positive samples as $A(x^p)$, the average embedding representation of negative samples as $A(x^n)$ and the average embedding representation of simulated samples as $A(x^s)$.

$$A(x^p) = \sum x^p/n \tag{3}$$

$$A(x^n) = \sum x^n/n \tag{4}$$

$$A(x^s) = \sum x^s/n \tag{5}$$

And the we define the margin-based triplet embedding regularization as follows:

$$R_{term} = max(0, m + \|x^p - x^n\|_2^2 - \|x^p - x^s\|_2^2) \tag{6}$$

In formula (6), the parameter m indicates how close the distance from the real sample to the simulated sample is than the distance from the real sample to the fake sample. We take the larger item of the distance difference and 0 as the regularization loss and add it to the neural network, to ensure that the Regularization loss will not negatively affect the classifier.

We find it is possible that we smooth the embedding space by jointly optimizing the original cross entropy and a large-margin based triplet distance constraint as a regularization term.

TABLE II
AUC (%) PERFORMANCE OF EACH METHOD ON DIFFERENT DATASETS.

	UADFV	Celeb-DF	DeepFakeDetection
Meso4[20]	83.22	65.82	49.45
MesoInception4	80.76	58.74	48.87
FWA[22]	97.40	54.10	71.66
EVA-MLP[16]	62.55	57.12	——
EVA-Log	62.16	57.45	
Multi-task[21]	68.09	35.45	50.54
Xception-c23[14]	88.56	68.57	79.58
Ours-CER	**99.99**	**99.95**	**98.43**

C. Networks

The Xception network is deep separable network, which reduces the parameters of the model. After data generation and data preprocessing, our model inputs samples into the Xception network for learning. Save the output representation of the last layer of the Xception networks for the calculation of the embedding space representation of the samples. Connect the softmax layer to the last layer and get the classification probability. As we mentioned before, the probability of the negative sample and the simulated sample are added together as the probability of the negative sample. During testing, we also add the probability of the negative sample and the simulated sample are added together as the probability of the negative sample. Finally, we reduced the three-classification problem to a two-classification problem. Add the regularization loss we defined earlier to the total loss with a coefficient λ . We set a smaller number of λ to ensure that it does not have a large impact. In our experiments, We set λ as 0.05. We use Adam optimization method and set learning rate starting from 0.001.

IV. EXPERIMENT

As shown in Figure 1, for positive and provided fake samples, we first extract each frame of the videos, and then perform face recognition on each frame and resize it to 224 * 224. Then we use the generation algorithm mentioned to get simulated samples. In order to ensure the randomness and diversity of the samples, we add a random size to the identified face rectangle, and then also resize it to 224 * 224 pixel. Generate the same size simulation sample 5 times for each picture as described above. We choose the Xception networks for training. Each iteration takes the same size of positive samples, negative samples, and simulated samples randomly. We selected three deepfake datasets, including UADFV, Celeb-DF, DeepFakeDetectio. Among them, UADFV is a low-quality video dataset, Celeb-DF and DeepFakeDetectio are high-quality video datasets. We also compared with several start-of-art open source deepfake detection algorithms. On the one hand, we conducted experiments on three datasets(UADFV/Celeb-DF/DeepfakeDetection) using open source algorithms provided and compared them with our own models under the same experimental conditions. At the same time, we also adjusted some parameters in our model to achieve better results, including different conditions, margin

TABLE III
AUC (%) PERFORMANCE OF OUR METHOD WITH DIFFERENT CONDITION INCLUDING CLASS NUMBER AND REGULARIZATION LOSS.

Different Condition/Dataset	DeepfakeDetection
Class Number = 3, add regularization loss	**9**8.17
Class Number = 3, not add regularization loss	98.03
Class Number = 2, not add regularization loss	80.85

TABLE IV
AUC (%) PERFORMANCE OF OUR METHOD WITH DIFFERENT PARAMETERS INCLUDING MARGIN VALUE AND BATCH SIZE.

Margin/batch size	4	8	16	24
10	97.49	97.47	98.31	98.23
20	97.68	97.68	98.06	98.17
30	97.32	97.91	98.35	98.15
40	97.09	98.02	**98.43**	98.09

value and batch size value. Detailed experimental results are discussed below.

A. Experiments On Different Datasets

We performed experiments on three datasets: UADDV, Celeb-DF, and DeepFakeDetection. The UADDV dataset is generated by the DeepFake algorithm including 49 real videos and corresponding generated fake videos. The Celeb-DF dataset is generated using a refined synthesis algorithm that reduces the visual artifacts observed in previous datasets including 408 real videos and 795 fake videos. To make the DeepFakeDetection dataset, google's researchers worked with paid and consenting actors to record 363 videos. Using publicly available deepfake generation methods, they then created 3068 deepfakes from these videos. For the UADFV and DeepFakeDetection datasets, we randomly select 70% of them as the training set and the remaining 30% as the test set. For the Celeb-DF dataset, we use the already divided training and test sets in the existing work [18]. Table I shows the detailed division of the datasets.

As shown in Table II, our model has achieved good results on all three datasets.

B. Compare with start-of-art

We chose some open source deepfake detection algorithms for comparison. Firstly, We briefly introduce these start-of-art algorithms.

- The Meso4 algorithm [33] proposes to detect forged videos of faces by replacing their method at a mesoscopic level of analysis. They think that microscopic analyses based on image noise cannot be applied in a compressed video context where the image noise is strongly degraded and at a higher semantic level, human eye struggles to distinguish forged images, especially when the image depicts a human face. The author replaces the first two convolutional layers of Meso4 with "initial modules" consisting of expanded convolutions to process multi-scale information called MesoInception4.
- We have introduced the FWA algorithm [37] in detail in the previous section. The author believes that the deepfake algorithm will produce some specific artifacts during the post-processing in order to adapt to different resolution videos. Deepfake detection can be performed by detecting these specific artifacts.
- The EVA algorithm [22] detects distinct visual features in tampered videos generated by different generation algorithms. Their model distinguish real videos from fake videos by detecting the reflection of missing incident light, and some details of eyes and teeth, detect Face2Face videos by detecting facial borders and nose features, detect generated face videos by find differences in color and position between eyes.
- The Multi-task algorithm [34] uses an auto-encoder with shared weights to locate tampered areas while detecting fake faces. Information is shared among the three subtasks, and the overall performance is improved by reducing the loss of each part.
- The last algorithm Xception-c23 [20], uses the Xception networks based on separable convolutions with residual connections and transfer it by replacing the final fully connected layer with two outputs. The author believes that adding additional domain-specific knowledge, such as face tracking method, can optimize the performance of the model.

We compared our model with these start-of-art deepfake detection algorithms. As mentioned above, We conducted experiments on different datasets using the provided open source algorithm. The universality of the FWA algorithm determines that it can be used without retraining on a new dataset. We use the training code provided by the EVA algorithm to retrain on its model. Table II shows the experimental results of these above-mentioned start-of-art algorithms on three datasets. It can be seen that our model has better results on all three datasets. We can also see that these algorithms perform better on the UADFV dataset than on the other two datasets, while the performance on the Celeb-DF dataset and DeepfakeDetection dataset has dropped significantly. This also confirms that with the continuous improvement of video tampering technology and the improvement of video quality, it does bring certain challenges to deepfake detection. Our model has also been affected to some extent, but it still has the better effect.

C. Experiments with Different Parameters for Our Model

To obtain better parameter configuration and verify the validity of our model, we also do some other comparative experiments.

We conducted experiments in three cases on DeepfakeDetection dataset. Firstly, we follow the method mentioned in section III, use three types of samples(real/fake/simulated samples) as input to the neural network, and add the regularization loss to the network. Then we only use two types of samples(real/fake samples) as input to the neural network, and add the regularization loss to the network. At last, we also use two types of samples(real/fake samples) as input to the neural network and not add the regularization loss. The other

experimental parameters are the same, and the batch size is set to 16, use Adam optimization method and set learning rate starting from 0.001.

As shown in Table III, we can see that we get the better result with using three types of samples and adding the regularization loss, and auc score is 98.32%. We can also find that using three types of samples get a much better result than using two types of samples with same other conditions.

We also do a series of comparative experiments by setting different parameter, including margin value and batch size. We take margin value of 10, 20, 30, and 40, and batch size value of 4, 8, 16, and 24, respectively. The other experimental conditions are the same. Using the Adam optimizer, the initial value of the learning rate is 0.001. Table IV shows the experimental results. We can see such a trend. As the batch size value increases, the experimental effect improves to some extent, and as the Margin value increases, the experimental results do not have a clear upward trend but there is still a certain improvement in general, but to some extent, it improves the performance of the model. We can see that the biggest auc score is 98.43%.

V. CONCLUSION

In this work, we propose a new method with clustering-based embedding regularization for deepfake detection. Our model achieved good results on UADFV, Celeb-DF and DeepFakeDetection datasets including low-quality videos and high-quality videos. And our experiments, show that the improvements we have made have significantly improved the results of deepfake detection. At the same time, video forgery technology is still improving, and video quality is still improving, we will continue improve our deepfake detection model.

ACKNOWLEDGMENT

This work is supported by the National Key Research and Development Program of China (2018YFC0831500), the National Natural Science Foundation of China under Grant No.61972047 and the NSFC-General Technology Basic Research Joint Funds under Grant U1936220.

REFERENCES

[1] "Deepfake algorithm," https://github.com/deepfakes/faceswap/, accessed February 8, 2020.
[2] "Faceswap algorithm," https://github.com/MarekKowalski/FaceSwap/, accessed February 8, 2020.
[3] "fakeapp algorithm," https://www.fakeapp.com/, accessed February 8, 2020.
[4] "faceapp algorithm," https://www.faceapp.com/, accessed February 8, 2020.
[5] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 2089–2093.
[6] R. Huang, S. Zhang, T. Li, and R. He, "Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2439–2448.
[7] Y. Lu, Y.-W. Tai, and C.-K. Tang, "Conditional cyclegan for attribute guided face image generation," *arXiv preprint arXiv:1705.09966*, 2017.
[8] L. Li, J. Bao, H. Yang, D. Chen, and F. Wen, "Faceshifter: Towards high fidelity and occlusion aware face swapping," *arXiv preprint arXiv:1912.13457*, 2019.
[9] P. Neekhara, S. Hussain, M. Jere, F. Koushanfar, and J. McAuley, "Adversarial deepfakes: Evaluating vulnerability of deepfake detectors to adversarial examples," *arXiv preprint arXiv:2002.12749*, 2020.
[10] A. Gandhi and S. Jain, "Adversarial perturbations fool deepfake detectors," *arXiv preprint arXiv:2003.10596*, 2020.
[11] N. Carlini and H. Farid, "Evading deepfake-image detectors with white- and black-box attacks," *arXiv preprint arXiv:2004.00622*, 2020.
[12] "Deepfakedetection dataset," https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html, accessed February 8, 2020.
[13] "facebook deepfake detection dataset," https://deepfakedetectionchallenge.ai/, accessed February 8, 2020.
[14] R. Tolosana, S. Romero-Tapiador, J. Fierrez, and R. Vera-Rodriguez, "Deepfakes evolution: Analysis of facial regions and fake detection performance," *arXiv preprint arXiv:2004.07532*, 2020.
[15] P. Korshunov and S. Marcel, "Vulnerability assessment and detection of deepfake videos," in *The 12th IAPR International Conference on Biometrics (ICB)*, 2019, pp. 1–6.
[16] B. Zhu, H. Fang, Y. Sui, and L. Li, "Deepfakes for medical video de-identification: Privacy protection and diagnostic information preservation," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 414–420.
[17] 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)*. IEEE, 2019, pp. 8261–8265.
[18] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: A new dataset for deepfake forensics," *arXiv preprint arXiv:1909.12962*, 2019.
[19] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *arXiv preprint arXiv:1812.08685*, 2018.
[20] 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.
[21] Y. Li, M.-C. Chang, and S. Lyu, "In ictu oculi: Exposing ai created fake videos by detecting eye blinking," in *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2018, pp. 1–7.
[22] 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)*. IEEE, 2019, pp. 83–92.
[23] L. Guarnera, O. Giudice, and S. Battiato, "Deepfake detection by analyzing convolutional traces," *arXiv preprint arXiv:2004.10448*, 2020.
[24] D. Güera and E. J. Delp, "Deepfake video detection using recurrent neural networks," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.
[25] E. Sabir, J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan, "Recurrent convolutional strategies for face manipulation detection in videos," *Interfaces (GUI)*, vol. 3, p. 1, 2019.
[26] X. Li, K. Yu, S. Ji, Y. Wang, C. Wu, and H. Xue, "Fighting against deepfake: Patch&pair convolutional neural networks (ppcnn)," in *Companion Proceedings of the Web Conference 2020*, 2020, pp. 88–89.
[27] N. Ruiz and S. Sclaroff, "Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems," *arXiv preprint arXiv:2003.01279*, 2020.
[28] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: A deepfake detection method using audio-visual affective cues," *arXiv preprint arXiv:2003.06711*, 2020.
[29] D. M. Montserrat, H. Hao, S. Yarlagadda, S. Baireddy, R. Shao, E. Bartusiak, J. Yang, D. Guera, F. Zhu, E. J. Delp *et al.*, "Deepfakes detection with automatic face weighting," *arXiv preprint arXiv:2004.12027*, 2020.
[30] R. Vignesh K *et al.*, "Deepfake video forensics based on transfer learning," *arXiv*, pp. arXiv–2004, 2020.
[31] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake video detection through optical flow based cnn," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
[32] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-stream neural networks for tampered face detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1831–1839.
[33] 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)*. IEEE, 2018, pp. 1–7.

[34] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," *arXiv preprint arXiv:1906.06876*, 2019.
[35] A. Kumar and A. Bhavsar, "Detecting deepfakes with metric learning," *arXiv preprint arXiv:2003.08645*, 2020.
[36] S. J. Sohrawardi, A. Chintha, B. Thai, S. Seng, A. Hickerson, R. Ptucha, and M. Wright, "Poster: Towards robust open-world detection of deepfakes," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2613–2615.
[37] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," *arXiv preprint arXiv:1811.00656*, 2018.
[38] Y. Zhong and W. Deng, "Adversarial learning with margin-based triplet embedding regularization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6549–6558.