Hindawi
Security and Communication Networks
Volume 2021, Article ID 9942754, 8 pages
https://doi.org/10.1155/2021/9942754

Research Article

FeatureTransfer: Unsupervised Domain Adaptation for Cross-Domain Deepfake Detection

Baoying Chen [1,2,3,4] **and Shunquan Tan** [1,2,3,4]

[1]*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China*
[2]*Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen, China*
[3]*Shenzhen Key Laboratory of Media Security, Shenzhen, China*
[4]*Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China*

Correspondence should be addressed to Shunquan Tan; tansq@szu.edu.cn

Received 16 March 2021; Revised 1 May 2021; Accepted 19 May 2021; Published 7 June 2021

Academic Editor: Mamoun Alazab

Recently, various Deepfake detection methods have been proposed, and most of them are based on convolutional neural networks (CNNs). These detection methods suffer from overfitting on the source dataset and do not perform well on cross-domain datasets which have different distributions from the source dataset. To address these limitations, a new method named FeatureTransfer is proposed in this paper, which is a two-stage Deepfake detection method combining with transfer learning. Firstly, The CNN model pretrained on a third-party large-scale Deepfake dataset can be used to extract the more transferable feature vectors of Deepfake videos in the source and target domains. Secondly, these feature vectors are fed into the domain-adversarial neural network based on backpropagation (BP-DANN) for unsupervised domain adaptive training, where the videos in the source domain have real or fake labels, while the videos in the target domain are unlabelled. The experimental results indicate that the proposed method FeatureTransfer can effectively solve the overfitting problem in Deepfake detection and greatly improve the performance of cross-dataset evaluation.

1. Introduction

Recently, the Deepfake video generation technology has attracted much attention, especially the popular Deepfake application called "ZAO". The application requires the user to provide a clear personal face image and complete facial feature verification, but the image collection protocol is not user-friendly. The majority of users express anxiety about the security of face information. In addition, the Deepfake technology could also be used to create fake news, posing threats to user privacy and social security [1–6]. Thus, it is critical to detect the Deepfake images or videos for face forensics. As we know, Deepfake detection, a branch of face forensics, is a binary classification task. The goal of face forensics is to detect whether a face in image or video has been created or manipulated.

The Deepfake video detection method mainly uses deep learning technology, which is usually composed of two parts: face detection and classification. As for face detection [7–9], MTCNN (multitask convolutional neural network) [7] and dlib [8] are mostly used as face detectors. As for the classification part, some researchers detect the Deepfake videos with the visible artifacts in the videos. For example, Matern et al. [10] found the inconsistent color of the left and right eyes and the geometric deformations of teeth in Deepfake videos. Li et al. [11] found that the people in Deepfake videos blink less frequently. Yang et al. [12] detected videos Deepfake through the cue of inconsistent head poses. Li et al. [13] exposed Deepfake videos by detecting face warping artifacts. These methods are effective for detecting some early Deepfake videos. However, with the development of Deepfake video generation technology, the visible artifacts used by these methods can be significantly reduced, degrading the performance of some artifacts-based methods. Therefore, some other cues in Deepfake videos need to be found for detection. Zhang et al. [14] found that the upsample or transposed

convolution used by the Deepfake technology inevitably results in a checkerboard effect on the generated face. Based on this, they proposed that CNN can be used to learn the checkerboard effect characteristics to detect Deepfake videos by directly inputting the face images extracted from video frames, such as MesoNet [15] and XceptionNet [16]. Unlike the spatial cues mentioned above, the temporal flickering, i.e., inconsistent temporal changes in videos, can be taken as the temporal cues in Deepfake videos. To make full use of both spatial and temporal cues in Deepfake videos, Guera et al. [17] and Chen et al. [18] combined CNN and recurrent neural networks (RNNs) to detect Deepfake videos. Unfortunately, Li et al. [19] found that most of the Deepfake detection methods trained and tested on specific datasets can achieve satisfactory performance, but their performances are significantly reduced when the methods are tested on cross-domain datasets, indicating that these methods are overfitting on a specific dataset. To improve the generalization ability of the methods on cross-domain datasets, multitask learning approaches [20–22] were introduced for Deepfake detection. Specifically, Nguyen et al. [20] developed a multitask learning approach to simultaneously perform classification, reconstruction, and segmentation of manipulated facial images. Cozzolino et al. [21] proposed the "ForensicTransfer" by combining classification and reconstruction, while Li et al. [22] proposed the "Face X-Ray" to detect Deepfake videos based on blending boundaries by combining classification and segmentation. However, those methods still need to improve the performance of the cross-dataset evaluation because they tend to train the classifier on a single small-scale dataset (i.e., FaceForensics++ [16] dataset), which is difficult to be generalized to other unseen datasets generated by using unseen Deepfake manipulation methods.

To make the Deepfake video detection method more robust on cross-domain datasets, this paper proposes a new method called FeatureTransfer, which is based on unsupervised domain adaptation. Extensive experiments demonstrate that the proposed method FeatureTransfer can improve the Deepfake detection performance of cross-dataset evaluation. The contributions of this work are summarized as follows:

(1) The unsupervised domain adaptation is first used to detect Deepfake videos in this work. A two-stage training pipeline called FeatureTransfer is designed for Deepfake detection.

(2) The feature extractor in preprocessing stage is pretrained on a large-scale Deepfake dataset DFDC-P [23] to extract more transferable feature vectors.

(3) Based on BP (backpropagation) and DANN (domain-adversarial neural network), an unsupervised domain adaptive network called BP-DANN is proposed.

The remainder of this paper is organized as follows. In Section 2, the related works are presented. In Section 3, our proposed method is described in detail. In Section 4, we provide comprehensive experimental results and analysis, as well as ablation studies. Finally, concluding remarks are drawn in Section 5.

2. Related Work

While the main focus of our work lies in the field of Deepfake detection, FeatureTransfer also intersects with the field of transfer learning, especially unsupervised domain adaptation. In the section, we clearly review previous Deepfake detection methods and transfer learning methods.

2.1. Deepfake Detection. To detect the Deepfake images or videos, most of the previous works are based on deep learning methods, which can be categorized into two detection methods: CNN-based methods [10, 13, 15, 16, 20–22] and RCNN-based methods [11, 17, 18]. The CNN-based methods extract face images from video frames and input them into the CNN for training and prediction to obtain the image-level result. These methods only use spatial information of a single frame in Deepfake videos. In addition, Qian et al. [24] detected Deepfake videos by mining clues in the frequency domain instead of the RGB domain. By contrast, the RCNN-based methods need a sequence of video frames for training and prediction to obtain the video-level result. These methods use both CNN and RNN, and they are called RCNN. Therefore, the RCNN-based methods can make full use of spatial and temporal information of Deepfake videos. Moreover, some Deepfake detection methods [12, 25] are based on traditional machine learning methods, Yang et al. [12] and Ciftci et al. [25] used SVM (support vector machine) as a classifier by extracting handcrafted features, such as biological signals. Finally, the methods mentioned above are summarized in Table 1.

2.2. Transfer Learning and Domain Adaptation. Transfer learning is an important branch of deep learning, which uses the knowledge of the source domain to assist the model in learning the knowledge of the target domain faster and better. Recently, transfer learning has been widely used in the field of forensics [21, 26, 27]. For example, loading the pretrained weight of ImageNet to the model before the model is trained is a simple transfer learning. Cozzolino et al. [21] trained the ForensicTransfer on the samples from the source domain and then performed fine-tuning with a small number of samples from the target domain to improve the performance of the ForensicTransfer on the target domain.

As a key field in transfer learning, domain adaptation aims to make the distribution of the source domain and the target domain in the feature space as close as possible. Meanwhile, the target model trained in the source domain can be transferred to the target domain to obtain good performance. Most works exploiting deep domain adaptation are based on discrepancy measurement. For instance, correlation alignment (CORAL) [28] and maximum mean discrepancy (MMD) [29] are used to reduce the distribution divergence between domains. Some works are based on discrepancy measurement domain-adversarial learning, such as domain-adversarial neural network (DANN) [30], multiadversarial domain adaptation (MADA) [31], and

Table 1: A summary of Deepfake detection methods.

Method	Classifier	Description
Matern et al. [10]	CNN	Handcrafted
Li et al. [13]	CNN	Self-supervised
MesoNet [15]	CNN	RGB
XceptionNet [16]	CNN	RGB
Nguyen et al. [20]	CNN	Multitask
ForensicTransfer [21]	CNN	Multitask
Face X-Ray [22]	CNN	Multitask
Qian et al. [24]	CNN	Frequency
Li et al. [11]	CNN + LSTM	Handcrafted
Guera et al. [17]	CNN + LSTM	RGB
Chen et al. [18]	CNN + LSTM	RGB
Yang et al. [12]	SVM	Handcrafted
FakeCatcher [25]	SVM	Handcrafted

transfer learning with dynamic adversarial adaptation network (DAAN) [32].

FeatureTransfer is a CNN-based method. In this work, a third-party Deepfake dataset is first used to train the CNN to extract the feature vectors of the face images. Then, the domain-adversarial neural network based on backpropagation (BP-DANN) is exploited for feature transfer training, which can improve the performance of Deepfake on cross-domain datasets.

3. Proposed Method

In this section, we introduce the details of the proposed method FeatureTransfer. Unlike the end-to-end adversarial training method NANN, FeatureTransfer exploits a two-stage adversarial training pipeline. As shown in Figure 1, the FeatureTransfer is composed of two parts: (a) the preprocessing stage, including face detection and feature vector extraction, and (b) BP-DANN unsupervised domain adaptive module.

3.1. Motivation. Most of the methods studying cross-dataset evaluation mainly trained the model on the FaceForensics++ [16] dataset or other small-scale datasets and then tested it on other datasets. Unfortunately, the methods used to generate Deepfake videos on different datasets are often different, which may lead to great gaps in the generated videos. As a result, it is difficult to train a model with good detection ability for all or most of the Deepfake datasets on a specific small-scale Deepfake dataset. In addition, many forensics methods are data-driven, so it is important to find a large-scale training model of the Deepfake dataset which contains a variety of Deepfake generation methods. Fortunately, a large-scale Deepfake dataset DFDC-F [23], including 23654 real videos and 104500 fake videos, meets our data-driven requirements. The fake videos in the DFDC-F dataset were created by different methods, including Deepfake Autoencoder (DFAE) [33], MM/NN face swap [34], NTH [35], and FSGAN [36]. Thus, the feature extractor CNN pretrained on the DFDC-F dataset can be used to extract more transferable feature vectors, which will be fed into BP-DANN for unsupervised domain adaptive training.

3.2. Problem Definition. In the unsupervised domain adaptation for Deepfake detection, it is assumed that the source distribution is $D_s = \{(x, y)|x \in X^s, y \in Y^s\}$, where X^s and Y^s are the input and label space of the source domain, respectively. Meanwhile, the target distribution is $D_t = \{(x, y)|x \in X^t, y \in Y^t\}$, where X^t and Y^t are the input and label space of the target domain. However, the input samples in the source domain are labelled but unlabelled in the target domain. D_s and D_t have the same label space so that $Y^s = Y^t = \{0, 1\}$, where "0" represents the real image or video and "1" represents the fake image or video. Moreover, each input x, the feature vector extracted from CNN in the preprocessing stage, has a domain label $d = 0$ if $x \in X^s$ while $d = 1$ if $x \in X^t$. The distributions between the two domains are similar, i.e., $D_s \cap D_t \neq \varnothing$ and $D_s \neq D_t$. This work aims to extract the more generalized feature vectors from the pretrained CNN in the preprocessing stage and design a deep neural network that enables learning of transferable features $f = G_f(x)$ and adaptive classifier $y = G_y(f)$ to reduce the gap between the two domains, such that the target risk $E_{(x,y)\sim D_t}[G_y(G_f(x)) \neq y]$ can be bounded by minimizing the source risk and the cross-domain discrepancy.

3.3. Preprocessing Stage. In the preprocessing stage, the face detection network MTCNN is first used to obtain the face region of the video frame, and the region is expanded by 1.2 times to crop the face image and save it. Then, the CNN (i.e., se_resnext101_32 × 4 d [37]) is pretrained on the third-party large-scale Deepfake dataset (i.e., DFDC-F [23]). Finally, the face images are fed into the CNN to extract the feature vectors with 2048 dimensions. The extracted feature vectors are saved so that they can be quickly loaded to the BP-DANN for unsupervised domain adaptive training.

3.4. Domain-Adversarial Network. The DANN can learn domain-invariant features through end-to-end adversarial training. The learning procedure is a two-player game: the first player is the domain discriminator G_d that is trained to distinguish the source domain from the target domain; the second player is the feature extractor G_f which extracts domain-invariant features that can confuse the domain discriminator. In the adversarial training for the two players, the parameter θ_f of feature extractor G_f is learned by maximizing the loss of the domain discriminator G_d, while the parameter θ_d of domain discriminator G_d is learned by minimizing the loss of the domain discriminator. In addition, the loss of label classifier G_y is also minimized. The overall loss function of DANN can be formalized as

$$L(\theta_f, \theta_y, \theta_d) = \frac{1}{n_s} \sum_{x_i \in D_s} L_y\left(G_y\left(G_f\left(x_i; \theta_f\right); \theta_y\right), y_i\right) - \frac{\lambda}{n_s + n_t} \sum_{x_i \in \left(D_s \cup D_t\right)} L_d\left(G_d\left(G_f\left(x_i; \theta_f\right); \theta_d\right), d_i\right), \tag{1}$$

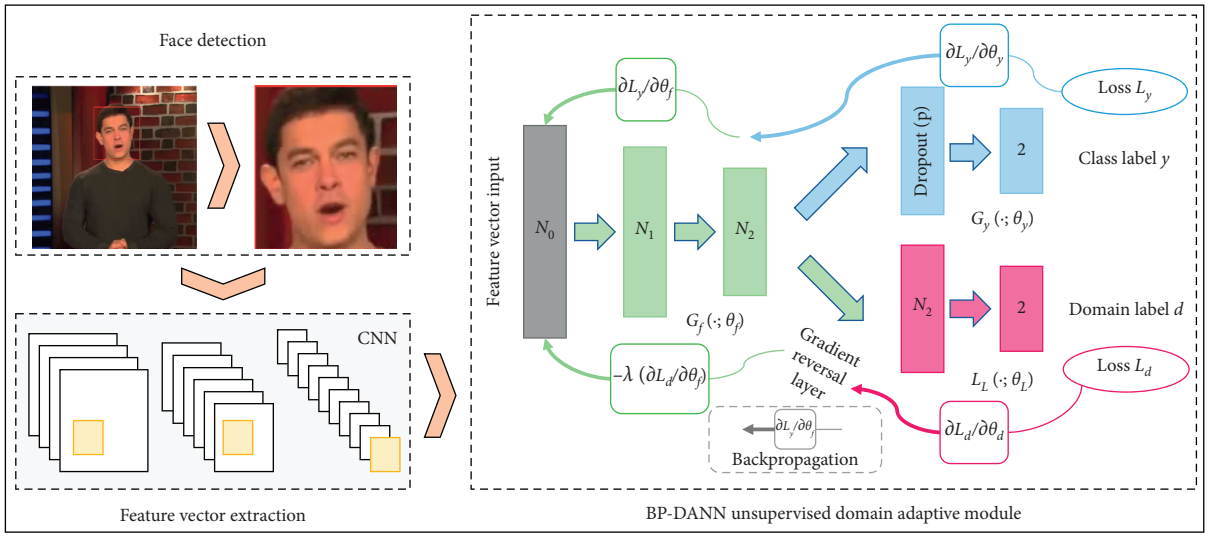

Figure 1: The pipeline of the proposed method FeatureTransfer. In the preprocessing stage, we obtain the face images of the video frame from the source and target domain and then feed them into CNN to extract the feature vectors. In the unsupervised domain adaptation stage, the BP-DANN consists of a feature extractor G_f (green), a label classifier G_y (blue), and a domain discriminator G_d (red). The gradient reversal layer connects G_f and G_d to realize unsupervised domain adaptation, and it multiplies the gradient by a certain negative constant during the backpropagation-based training.

where n_s and n_t are the number of samples in the source domain and the target domain, respectively, $d_i \in \{0, 1\}$ is the domain label of x_i, L_y is the loss for label prediction while L_d is the loss for domain discriminator, and λ is a hyper-parameter to trade-off the label classifier and the domain discriminator in the optimization problem. Based on equation (2) and equation (3), the optimization problem is to find the optimal parameters $\widehat{\theta}_f$, $\widehat{\theta}_y$, and $\widehat{\theta}_d$ that deliver a saddle point of equation (1) after the training converges.

$$\left(\widehat{\theta}_f, \widehat{\theta}_y\right) = \arg\min_{\theta_f, \theta_y} L\left(\theta_f, \theta_y, \widehat{\theta}_d\right), \tag{2}$$

$$\left(\widehat{\theta}_d\right) = \arg\min_{\theta_d} L\left(\theta_f, \theta_y, \widehat{\theta}_d\right). \tag{3}$$

3.5. BP-DANN Network Architecture. As shown in Figure 1, the network architecture of the proposed BP-DANN consists of three parts: feature extractor G_f, label classifier G_y, and domain discriminator G_d. These three parts are built by BP structure. G_f is composed of two fully connected layers, i.e., $L_f(N_0, N_1)$ and $L_f(N_1, N_2)$. The input and output dimensions of $L_f(N_0, N_1)$ are N_0 and N_1, where N_0 is 2048 and N_1 is 512. N_2 in $L_f(N_1, N_2)$ is set as 64. G_y is composed of a dropout layer with probability (p) of 0.5 and a fully connected layer $L_y(N_2, 2)$. G_d is composed of two fully connected layers, i.e., $L_d(N_2, N_2)$ and $L_d(N_2, 2)$. To obtain the more appropriate values of N_1, N_2, and p, the grid search is used for traversal search in this work.

4. Experiment

4.1. Dataset. In this section, the datasets related to the experiment are first introduced. Then, the details of the experiment implementation are given, and the experimental results are finally analyzed.

The DeepfakeTIMIT (DF-TIMIT) [38] dataset contains 640 Deepfake videos generated with a GAN-based method [39] and based on VidTIMIT [40] dataset. The videos are divided into two equal subsets: lower quality (LQ) and higher quality (HQ). In our experiment, we add 320 real videos of 32 related subjects in VidTIMIT, and the LQ subset is used for test.

The FaceForensics++ (FF) [16] dataset contains 1000 pristine (P) videos and 4000 fake videos generated by using the four most advanced facial manipulation methods, including DeepFakes (DF), Face2Face (F2F), FaceSwap (FS), and NeuralTextures (NT). This dataset covers three versions of compression qualities: Raw, c23, and c40. In our experiment, the FF-DF and FF-FS subsets with a compression quality of c23 are taken.

The DeepFakeDetection (DFD) [41] contains 363 real videos and 3068 Deepfake videos released by Google. Similar to FF, this dataset also covers three versions of compression qualities, including Raw, c23, and c40. In our experiment, c23 is taken.

The Celeb-DF [19] includes 408 real videos and 795 synthesized videos generated by using an improved version of the Deepfake algorithm.

The DFDC [23] dataset contains two versions: DFDC-Preview (DFDC-P) [42] and DFDC-Final (DFDC-F) [23]. The DFDC-P includes 1131 real videos and 4113 fake videos. The DFDC-F was released for the Deepfake Detection Challenge, and it includes 23654 real videos and 104500 fake videos. In our experiment, DFDC-F is taken to pretrain the CNN (i.e., se_resnext101_32 × 4 d), and DFDC-P is used for test.

As mentioned above, 30 frames are extracted from each video at equal intervals. Then, the face region of each frame is detected and saved as a face image. To balance the real and fake face images in DFDC-F, 30 frames from each fake video are extracted, but 150 frames from each real video are extracted. The numbers of face images in each dataset are listed in Table 2.

4.2. Implementation Details. Unlike the end-to-end adversarial learning training in DANN, a two-stage training strategy is adopted for FeatureTransfer.

In the first stage, a large-scale Deepfake dataset DFDC-F is used to train the CNN (i.e., se_resnext101_32 × 4 d). The CNN was initialized with pretrained weights on ImageNet, such that it can be used to extract more transferable feature vectors. The batch size is set to 128, and the total training epoch is 10. The Adam optimizer is used, where the initial learning rate is set to 2×10^{-3} and weight decay of 4×10^{-5}. After training, the CNN is used to extract the feature vectors of images, and the feature vectors are saved according to different datasets.

In the second stage, the feature vectors are loaded, and the BP-DANN is then trained. During the unsupervised domain adaptive adversarial training, the feature vectors of FF-DF (train set) are selected as the source domain, while the feature vectors of other test datasets are selected as the target domain. It should be noted that, due to a large number of images in the DFD, DFDC-P, and Celeb-DF datasets, only 10% of the images (the number of real and fake images is the same) in each dataset are used as the target domain for unsupervised adversarial training, and all images in each dataset are then tested after training. As for FF-FS and DF-TIMIT datasets, all images in the datasets are used as the target domain for unsupervised adversarial training, where the batch size is set to 128 and the total training epoch is 50. Instead of SGD used in DANN, the Adam optimizer with an initial learning rate of 1×10^{-4} is used. To suppress noisy signals from the domain classifier at the early stages of the training procedure, the hyperparameter λ in equation (1) is changed from 0 to 1 gradually based on the following equation:

$$\lambda = \frac{2}{1 + \exp(-\gamma \times p)} - 1, \tag{4}$$

where p is the training progress linearly changing from 0 to 1 and γ is set to 10.

4.3. Results and Analysis. The proposed method is compared with previous Deepfake detection methods, including Xception [16], FSSpotter [18], Face X-Ray [22], and se_resnext101_32 × 4 d [37]. The cross-domain Deepfake detection results are exhibited in terms of AUC (area under the curve) and ERR (equal error rate) on recently released datasets, such as DF-TIMIT, FF-FS (test set), DFD, DFDC-P, and Celeb-DF. The pretrained weight (all c23. p) provided by the author is loaded into Xception, and the model is then directly used to test on other datasets without retraining. Similarly, the se_resnext101_32 × 4 d is trained on DFDC-F, and the trained model is then directly used to test on other datasets without retraining. Due to the lack of open-source code for FSSpotter and Face X-Ray, the experimental results in the corresponding papers are directly used for comparison. The result with the clip length (T) of 1 in FSSpotter trained on FF-DF dataset is chosen as the image-level result. The Face X-Ray in the paper is trained on FF and BI [22] datasets.

Table 3 listed the cross-domain performance of all compared methods on different datasets. It can be seen that FeatureTransfer achieves the best performance on DFDC-P (seen dataset) and Celeb-DF (unseen dataset) compared to other methods in terms of AUC and ERR. Also, FeatureTransfer obtains a comparable result in FF-FS (unseen facial manipulations), DFD (unseen dataset), and DF-TIMIT (unseen dataset). In addition, Xception obtains the best performance on DF-TIMIT (unseen dataset) and FF-FS (seen dataset), while Face X-Ray obtains the best performance on DFD (unseen dataset) in terms of AUC and ERR. The performance of FSSpotter is relatively general, which could be caused by the fact that FSSpotter was only trained on the FF-DF dataset. However, the AUC result of the proposed method is only 2.24% lower than that of Xception on DF-TIMIT and 2.24% lower than that of Face X-Ray on DFD. Compared with se_resnext101_32 × 4 d, FeatureTransfer achieves a performance improvement ranged from 1% to 8% in terms of AUC on different datasets, especially 8% on the Celeb-DF. Compared with Xception, se_resnext101_32 × 4 d obtains better performance on more datasets, and this is why se_resnext101_32 × 4 d is used as the feature extractor of FeatureTransfer. In general, the results indicate that FeatureTransfer achieves better or comparable performance on cross-dataset evaluation, which mainly benefits from the more transferable feature vectors extracted from the deeper CNN called se_resnext101_32 × 4 d that was pretrained on a large-scale dataset DFDC-F. Moreover, using unsupervised domain adaptation can also improve the performance of the unlabelled Deepfake datasets in target domain.

4.4. Ablation Studies. To confirm the effectiveness of the proposed method, we explore the effect of different level evaluation and the effect of different training strategies in this section.

4.5. Effect of Different Level Evaluation. To verify the effectiveness and better generalization of the proposed method on different levels of evaluation, the results of image level and video level are compared. To get the video-level result, the prediction score for video is the predicted probability that the video is fake, which is calculated by averaging the scores of face images extracted from frames of a video. It can be seen from the image-level and video-level results shown

Table 2: The numbers of face images from each dataset.

	FF-DF		FF-FS		DF-TIMIT	DFD	Celeb-DF	DFDC-P	DFDC-F
	Train	Test	Valid	Test	LQ				
Real	21600	4200	4200	4200	9600	10890	12240	33897	2839521
Fake	21600	4200	4200	4200	9600	91740	23850	123412	2885045

Note. "Valid" is the short form of validation.

Table 3: The image-level results of all compared methods in terms of AUC (%) and EER (%) on each dataset.

Method	Test set									
	DF-TIMIT		FF-FS		DFD		DFDC-P		Celeb-DF	
	AUC	ERR	AUC	ERR	AUC	ERR	AUC	ERR	AUC	ERR
Xception [16]	98.80	5.95	99.56	2.74	83.06	25.92	82.10	27.23	72.54	34.71
FSS [18]	97.33	—	—	—	—	—	—	—	76.26	—
X-Ray [22]	—	—	98.00	-	95.40	8.37	80.92	27.54	80.58	26.70
Se_Res [37]	90.61	16.22	84.52	22.83	89.02	21.06	97.99	6.25	78.21	29.80
FT (ours)	96.56	8.05	88.62	19.52	91.00	16.21	98.77	5.75	86.21	22.42

Note. The "FSS," "X-Ray," "Se_Res," and "FT" are the short forms of "FSSpotter," "Face X-Ray," "se_resnext101_32 × 4 d," and "FeatureTransfer," respectively.

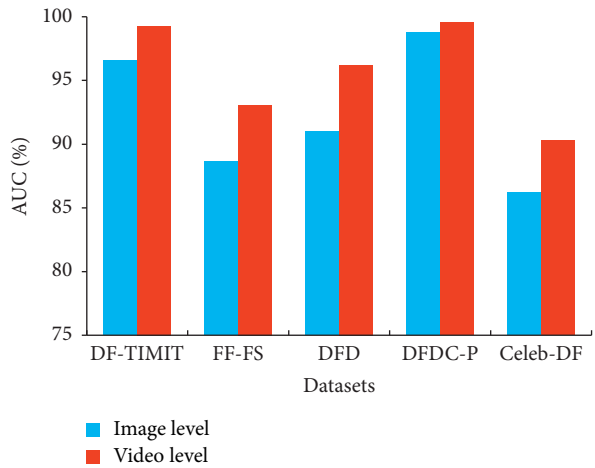

Figure 2: The results of different levels in terms of AUC (%) on each dataset.

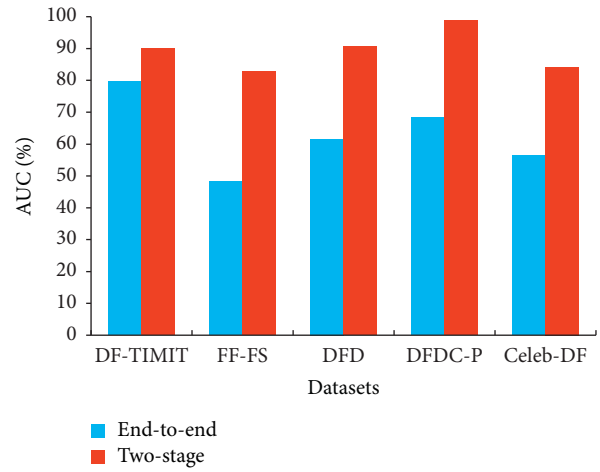

Figure 3: The image-level results of different training strategies in terms of AUC (%) on each dataset.

in Figure 2 that the video-level results are significantly improved on each dataset in terms of AUC (%).

4.6. Effect of Different Training Strategies. To demonstrate the benefits of the two-stage training strategy used in the proposed method, the experiments are conducted with the proposed FeatureTransfer and DANN having the same training epoch of 20. It should be noted that only the feature vectors of the source domain FF-DF (train set) and the target domain FF-FS (validation set) are used for unsupervised adversarial learning in our proposed method FeatureTransfer. The trained model is then directly evaluated on other datasets without additional adversarial learning. The backbone of DANN is se_resnext101_32 × 4 d, and DANN is trained by using an end-to-end training strategy with FF-DF (train set) as the source dataset and FF-FS (validation set) as the target dataset. As shown in Figure 3, in terms of AUC (%), the image-level results of FeatureTransfer using the two-stage training strategy are significantly improved on each dataset compared with DANN using the end-to-end training strategy.

5. Conclusions

In this work, FeatureTransfer, a two-stage Deepfake detection method based on unsupervised domain adaptation, is proposed. The feature vectors extracted from CNN are used for adversarial transfer learning in BP-DANN, which contributes to better performance than the end-to-end adversarial learning. Moreover, the feature extractor CNN pretrained on a large-scale Deepfake dataset can be used to extract more transferable feature vectors, which greatly reduce the gap between the source domain and the target domain during unsupervised domain adaptive training. The experimental results indicate that the proposed method achieves better and comparable performance for cross-domain Deepfake detection compared with previous methods.

However, there are still some limitations in our work. It is not an end-to-end detection method, and it needs a large-scale Deepfake dataset to pretrain the CNN to extract more transferable features, which takes a lot of time. Thus, in future work, we will devote ourselves to studying an end-to-end domain adaptive Deepfake detection method that does not require pretrained feature extractors.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Key-Area Research and Development Program of Guangdong Province (2019B010139003), NSFC (61772349, U19B2022, and 61872244), Guangdong Basic and Applied Basic Research Foundation (2019B151502001), and Shenzhen R&D Program (JCYJ20180305124325555). This work was also supported by Alibaba Group through Alibaba Innovative Research (AIR) Program.

References

[1] S. Hakak, M. Alazab, S. Khan, T. R. Gadekallu, P. K. R. Maddikunta, and W. Z. Khan, "An ensemble machine learning approach through effective feature extraction to classify fake news," *Future Generation Computer Systems*, vol. 117, pp. 47–58, 2021.

[2] S. Hakak, W. Z. Khan, S. Bhattacharya, G. T. Reddy, and K.-K. R. Choo, "Propagation of fake news on social media: challenges and opportunities," in *Proceedings of the International Conference On Computational Data And Social Networks*, pp. 345–353, Dallas, TX, USA, December 2020.

[3] M. A. Azad, M. Alazab, F. Riaz, J. Arshad, and T. Abullah, "Socioscope: I know who you are, a robo, human caller or service number," *Future Generation Computer Systems*, vol. 105, pp. 297–307, 2020.

[4] R. Sagar, R. Jhaveri, and C. Borrego, "Applications in security and evasions in machine learning: a survey," *Electronics*, vol. 9, no. 1, p. 97, 2020.

[5] A. R. Javed and Z. Jalil, "Byte-level object identification for forensic investigation of digital images," in *Proceedings of the 2020 International Conference On Cyber Warfare And Security (ICCWS)*, pp. 1–4, Norfolk, VA, USA, March 2020.

[6] A. Rehman, S. U. Rehman, M. Khan, M. Alazab, and T. Reddy, "CANintelliIDS: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU," *IEEE Transactions on Network Science and Engineering*, vol. 2021, Article ID 3059881, 1 page, 2021.

[7] X. Yin and X. Liu, "Multi-task convolutional neural network for pose-invariant face recognition," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 964–975, 2017.

[8] D. E. King, "Dlib-ml: a machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[9] H. Zhang, A. Jolfaei, and M. Alazab, "A face emotion recognition method using convolutional neural network and image edge computing," *IEEE Access*, vol. 7, pp. 159081–159089, 2019.

[10] F. Matern, C. Riess, and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proceedings of the 2009 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, pp. 83–92, IEEE, Snowbird, UT, USA, December 2009.

[11] Y. Li, M. Chang, and S. Lyu, "In ictu oculi: exposing ai created fake videos by detecting eye blinking," in *Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, IEEE, Hong Kong, China, December 2018.

[12] X. Yang, Y. Li, and S. Lyu, "Exposing deep fakes using inconsistent head poses," in *Proceedings of the ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8261–8265, IEEE, Brighton, UK, May 2019.

[13] Y. Li and S. Lyu, "Exposing deepfake videos by detecting face warping artifacts," 2018, https://arxiv.org/abs/1811.00656.

[14] X. Zhang, S. Karaman, and S. Chang, "Detecting and simulating artifacts in GAN fake images," in *Proceedings of the 2019 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–6, IEEE, Delft, Netherlands, December 2019.

[15] D. Afchar, V. Nozick, J. Yamagishi, and I. Echizen, "Mesonet: a compact facial video forgery detection network," in *Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, pp. 1–7, IEEE, Hong Kong, China, December 2018.

[16] A. Rossler, D. Cozzolino, L. Verdoliva et al., "Faceforensics++: learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1–11, Seoul, South Korea, October 2019.

[17] D. Güera, E. J. Delp, Deepfake video detection using recurrent neural networks," in *Proceedings of the 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–6, IEEE, Auckland, New Zealand, November 2018.

[18] P. Chen, J. Liu, T. Liang et al., "Fsspotter: spotting face-swapped video by spatial and temporal clues," in *Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, IEEE, London, UK, July 2020.

[19] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df: a large-scale challenging dataset for deepfake forensics," in *Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3207–3216, Salt Lake City, UT, USA, July 2020.

[20] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," 2019, https://arxiv.org/abs/1906.06876.

[21] D. Cozzolino, J. Thies, A. Rössler et al., "Forensictransfer: weakly-supervised domain adaptation for forgery detection," 2018, https://arxiv.org/abs/1812.02510.

[22] L. Li, J. Bao, T. Zhang et al., "Face x-ray for more general face forgery detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5001–5010, Seattle, WA, USA, June 2020.

[23] B. Dolhansky, J. Bitton, B. Pflaum et al., "The deepfake detection challenge dataset," 2020, https://www.arxiv-vanity.com/papers/2006.07397/.

[24] Y. Qian, G. Yin, L. Sheng, Z. Chen, and J. Shao, "Thinking in frequency: face forgery detection by mining frequency-aware

clues," in *Proceedings of the European Conference On Computer Vision*, pp. 86–103, Glasgow, UK, August 2020.

[25] U. A. Ciftci, I. Demir, and L. Yin, "Fakecatcher: detection of synthetic portrait videos using biological signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 2020, Article ID 3009287, 1 page, 2020.

[26] H. Lin, J. Hu, W. Xiaoding, M. F. Alhamid, and M. J. Piran, "Towards secure data fusion in industrial IoT using transfer learning," *IEEE Transactions on Industrial Informatics*, vol. 2020, Article ID 3038780, 1 page, 2020.

[27] R. Abbasi, A. Kashif Bashir, J. Chen et al., "Author classification using transfer learning and predicting stars in co-author networks," *Software: Practice and Experience*, vol. 51, no. 3, pp. 645–669, 2020.

[28] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation, Lecture Notes in Computer Science," in *Proceedings of the European Conference on Computer Vision*, pp. 443–450, Springer, Glasgow, UK, August 2016.

[29] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proceedings of the International Conference on Machine Learning*, pp. 97–105, PMLR, Long Beach, CA, USA, June 2015.

[30] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," ", PMLR, in *Proceedings of the International Conference on Machine Learning*, pp. 1180–1189, PMLR, Lille, France, July 2015.

[31] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," 2018, https://arxiv.org/abs/1809.02176.

[32] C. Yu, J. Wang, Y. Chen, and M. Huang, "Transfer learning with dynamic adversarial adaptation network," in *Proceedings of the 2019 IEEE International Conference On Data Mining (ICDM)*, pp. 778–786, IEEE, Beijing, China, November 2009.

[33] I. Petrov, D. Gao, N. Chervoniy et al., "Deepfacelab: a simple, flexible and extensible face swapping framework," 2020, https://arxiv.org/abs/2005.05535.

[34] D. Huang and F. De La Torre, "Facial action transfer with personalized bilinear regression," in *Proceedings of the European Conference on Computer Vision*, pp. 144–158, Florence, Italy, October 2012.

[35] E. Zakharov, A. Shysheya, E. Burkov, and V. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9459–9468, Seoul, Korea, October 2019.

[36] Y. Nirkin, Y. Keller, and T. Hassner, "FSGAN: Subject agnostic face swapping and reenactment," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7184–7193, Seoul, Korea, October 2019.

[37] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, June 2018.

[38] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," 2018, https://arxiv.org/abs/1812.08685.

[39] Shaoanlu, "faceswap-gan github," 2020, https://github.com/shaoanlu/faceswap-GAN.

[40] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference, Advances in Biometrics," in *Proceedings of the International Conference on Biometrics*, pp. 199–208, Springer, Alghero, Italy, June 2009.

[41] N. Dufour, Google Research, and J. A. Gully, "Contributing data to deepfake detection research," 2020, https://ai.googleblog.com/2019/09/contributing-data-to-deepfake-detection.html.

[42] B. Dolhansky, R. Howes, B. Pflaum, N. Baram, and C. C. Ferrer, "The deepfake detection challenge (DFDC) preview dataset," 2019, https://arxiv.org/abs/1910.08854.