Face Forgery Detection Based on Facial Region Displacement Trajectory Series

YuYang Sun[1,3], ZhiYong Zhang[3], Isao Echizen[1,2], Huy H. Nguyen[2], ChangZhen Qiu[3], and Lu Sun[4]
[1]The University of Tokyo, Japan [2]National Institute of Informatics, Japan
[3]Sun Yat-sen University, China [4]South China University of Technology, China
tarrysun0115@g.ecc.u-tokyo.ac.jp, {nhhuy,iechizen}@nii.ac.jp

Abstract

Deep-learning-based technologies such as deepfakes ones have been attracting widespread attention in both society and academia, particularly ones used to synthesize forged face images. These automatic and professional-skill-free face manipulation technologies can be used to replace the face in an original image or video with any target object while maintaining the expression and demeanor. Since human faces are closely related to identity characteristics, maliciously disseminated identity manipulated videos could trigger a crisis of public trust in the media and could even have serious political, social, and legal implications. To effectively detect manipulated videos, we focus on the position offset in the face blending process, resulting from the forced affine transformation of the normalized forged face. We introduce a method for detecting manipulated videos that is based on the trajectory of the facial region displacement. Specifically, we develop a virtual-anchor-based method for extracting the facial trajectory, which can robustly represent displacement information. This information was used to construct a network for exposing multidimensional artifacts in the trajectory sequences of manipulated videos that is based on dual-stream spatial-temporal graph attention and a gated recurrent unit backbone. Testing of our method on various manipulation datasets demonstrated that its accuracy and generalization ability is competitive with that of the leading detection methods.

1. Introduction

With the development and spread of high-resolution, and rich-diversity deep generation models such as generative adversarial networks (GANs) [13], images can be synthesized with sufficient texture details to fool the casual viewer. This not only illustrates the potential creativity of artificial intelligence but also lays bare the hidden dangers for digital information security. Deep-learning-based digital face manipulation technologies have gradually become widely used. These manipulation technologies can be used to maliciously edit, deform and replace faces in images or videos, which can then be used pornographically to damage reputations or to spread fake news and hate speech with a celebrity connection, causing political tensions and democratic crises. In addition, more and more open-source deepfakes applications such as FaceApp [1] and ZAO [2] have reduced the need for manual editing in video face swapping, so anyone with a laptop and Internet connection can easily synthesize and spread forged face videos.

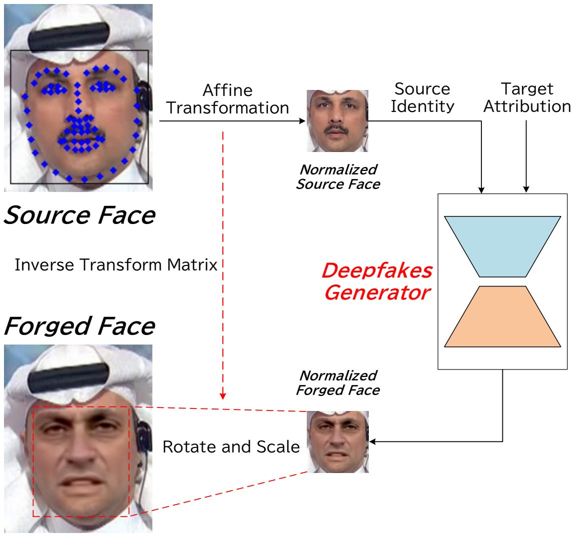

Figure 1. A brief overview of DeepFake generation pipeline. The synthesized normalized manipulated face utilizes the landmark information of the source face for inverse affine transformation to match the source video frame, which may cause spatial artifacts in local facial regions and further form inconsistent temporal abruptness in frame-by-frame manipulation.

Therefore, it is essential to propose robust detection methods to counteract the potential threats caused by the proliferation of synthetic and manipulated videos in order to protect personal privacy and property security. Current mainstream image-based detection methods only focus on exposing frame-level artifacts in the spatial or frequency domain and determine the video-level results through independent multi-frame detection, which ignores the temporal inconsistency information hidden in the video stream. We assert that manipulation leaves temporally unnatural marks in several spatial regions of the face. Therefore, simultaneously capturing the intra-frame spatial coherence between facial regions and the inter-frame temporal consistency of information in specific areas is of great help in identifying forged faces. Our method is dedicated to exposing subtle spatial artifacts and inconsistent temporal abruptness in critical facial regions caused by the forced blending of the synthesized normalized forged face in the deepfakes pipeline.

As the first step in developing a robust detection method, we statistically analyze the difference in landmark sequences between real and fake videos (Section 3) in order to determine whether digital face manipulation creates specific artifacts in the micro-motion trajectory of the face. Next, we propose a virtual-anchor-based method for extracting the facial region displacement trajectory (Section 4). Several local facial regions are taken as tracking targets, and the feature points with excellent tracking characteristics are screened in each region frame-by-frame to robustly and accurately obtain the displacement trajectory. Finally, we develop a fake trajectory detection network (FTDN) based on dual-stream spatial-temporal graph attention and a gated recurrent unit (GRU) backbone for detecting fake trajectories (Section 5). The network utilizes the extracted trajectory and explicitly aggregates the important information of different dimensions in the input sequences, which can help to effectively capture the spatial-temporal anomalies in the manipulated video trajectory. We tested our model on various manipulation techniques in the FaceForensics++ [27] dataset and achieved excellent detection performance.

We summarize our contributions as follows:

- We analyze and verify the impact of face forgery facial landmark series, and thus construct a DeepFakes detection method by capturing the spatial-temporal anomalies of facial region displacement trajectory;
- We design a robust facial region displacement trajectory extraction method based on virtual anchor to highlight the local displacement features;
- We propose a Fake Trajectory Detection Network (FTDN) to effectively identify forgery spatial-temporal patterns based on the facial region displacement trajectory with competitive results on the FaceForensics++ dataset.

2. Related Work

2.1. Deepfakes Detection

To identify synthetic face images, early researchers focused on capturing specific artifacts caused by defects of deepfakes generators. Well-known examples of such artifacts are inconsistent head poses [33], abnormal facial expressions and head movements [4], broken photo-response non-uniform patterns [20], and detectable differences based on image quality measures [21]. These artifacts may strongly affect certain deepfakes methods but do not generalize well, resulting in poor robustness. Subsequent work used neural networks to adaptively mine the high-level features of forged face images and learn the pattern differences [3, 25, 22], which could substantially improve model accuracy and generalizability. For video-level detection, researchers generally believe that low-level artifacts caused by face manipulation manifest as temporal artifacts that are inconsistent across the frame, so they often take temporal information into consideration. As representative examples, Sabir *et al.* [28] leveraged a bidirectional GRU [7] on the feature output of a convolutional neural network (CNN) backbone to identify the temporal patterns. Güera and Delp [14] used a CNN to extract frame-level features and fed them into a long short-term memory (LSTM) [15] to create temporal descriptors for classification.

In particular, previous work [11, 34, 4, 31] has shown the potential of using facial geometric information in face manipulation detection. However, these works did not simultaneously mine the temporal inconsistency of facial geometric features caused by frame-by-frame manipulation and the spatial information anomalies between different facial regions caused by the distortion in the process of forgery face synthesis. Inspired by the previous works, we choose to transform the task of detecting deepfakes videos into the task of detecting multi-variable time series anomalies to expose artifacts caused by facial manipulation in both temporal and spatial dimensions.

2.2. Graph Neural Networks

Graph neural networks (GNNs) are deep learning models based on a graph structure, in which the nodes and edges are used to structure the content and attributes of the target so as to improve the representation ability of non-Euclidean data. Scarselli *et al.* [29] first proposed the concept, i.e., the combining of graph data with a neural network to carry out end-to-end calculation on structured data. Bruna *et al.* subsequently proposed incorporating local feature extraction and weight sharing into the calculation of the graph structure, resulting in a graph convolutional network (GCN) [6].

Further studies revealed that the attention mechanism can help the neural networks effectively capture the essential features of the data and avoid redundant information. This led to Veličković *et al.* devising the graph attention network (GAT) [32], a special form of the spatial-based GCN. They introduced the attention mechanism into node aggregation, aiming to assign weights to neighbor nodes in accordance with their importance when aggregating the information of the target node. Brody *et al*. [5] improved the attention formula and proposed using dynamic GAT to further improve the effect of aggregation. Here we use the dynamic graph attention mechanism to explicitly add attention weights to the spatial-temporal trajectory sequence, which effectively helps our model to focus on the anomalies caused by face forgery.

2.3. Time Series Processing

A time series is a very common form of data and is generally a sequence of measured values obtained by sampling variables in time-series order. Traditional time-series processing uses an auto-regressive moving average model to fit the sequences. Advances in machine learning led to some researchers such as Kate *et al*. combining k-nearest neighbor classification with dynamic time warping [16] to robustly represent misaligned time-series data [18]. Subsequent work demonstrated that deep learning methods are effective for processing time-series data. For instance, Malhotra *et al*. devised a pre-trained deep recurrent neural network called "TimeNet" for classifying time-series data [24] that uses a recurrent neural network (RNN) to construct encoder-decoder pairs and to learn a pre-trained temporal feature extraction model in an unsupervised way. Karim *et al*. proposed using an LSTM-FCN architecture [17] to learn the local features and global correlation of time series data.

Some researchers have speculated that the correlation between features and between time steps in a multi-variable time series can be effectively represented by a graph structure and have combined GNNs with time series learning. For example, Zhao *et al*. proposed using a MTAD-GAT [35], which combines graph attention information into a time-series representation, to effectively highlight the anomalies in sequences. Similarly, Deng and Hooi [12] created a learnable graph structure that enables the time-series anomaly detection model to adaptively learn the relationships between feature nodes. Inspired by these ideas, we decided to use dual-stream spatial-temporal graph attention to identify as much as possible the abnormal artifacts in the trajectory sequences of manipulated videos.

3. Method Feasibility Analysis

Since a deepfakes generator requires input and output images of fixed size, the source face needs to be normalized. As shown in Figure 1, the landmarks in the source face are first extracted, and these feature coordinates are used to construct an affine transformation matrix, which is used to convert the source face into a normalized form with fixed size and head pose. The deepfakes generator then combines the identity information of the source face with the attribute information of the target face to synthesize a forged face. To ensure that the forged face perfectly matches the contour of the source face, the above matrix is used for inverse affine transformation of the normalized forged face. Analysis of this process led to the following conjecture: Due to the structural differences from person to person, using the source face's features to forcibly rotate and scale the manipulated output inevitably introduces spatial errors into local regions of the forged face, and further become capturable temporal artifacts in trajectory sequence during frame-by-frame processing. These artifacts could be helpful in deepfake detection.

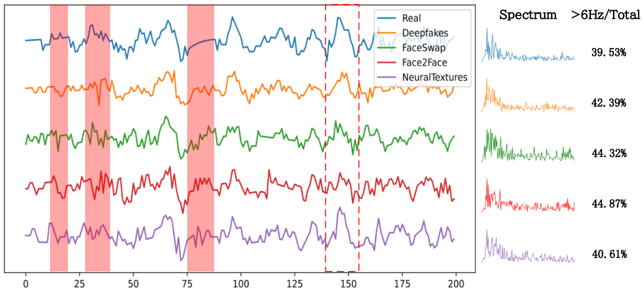

Figure 2. Landmark longitudinal trajectory diagram and corresponding spectrum for real video and several manipulated videos (series have been normalized and detrended).

To prove our conjecture, we extracted the temporal trajectories of the longitudinal position of a specific landmark in an original video and several corresponding manipulated versions. As shown in Figure 2, the trends of these trajectories are generally consistent; however, those for the forgeries are completely opposite to that for the real trajectory at three time steps (red-shaded areas). In addition, the trajectories of the forgeries have more burrs and spikes and are much coarser than that of the real trajectory (e.g., red-outlined box). The figure also shows the spectra of these trajectories and the proportion of high-frequency components ($> 6Hz/Total$). The real trajectory had the smallest proportion of high-frequency components (39.53%). The Deepfakes, FaceSwap, and Face2Face forgeries obviously had more high-frequency burr noise and higher proportions (42.39%, 44.32%, and 44.87%, respectively). Due to the textures rendering, NeuralTextures had a smaller fusion error, so its trajectory is similar to that of the real one, and its proportion of high-frequency components relatively low.

The high-frequency burrs are attributed to the spatial error caused by the abnormal warping of the forged face during the inverse affine transformation and are reflected in the abnormal movement trends of the feature points in the trajectories. This suggests that the facial region displacement trajectory is a robust feature with pattern differences and that its spatial-temporal information can be used to effectively expose the face manipulation.

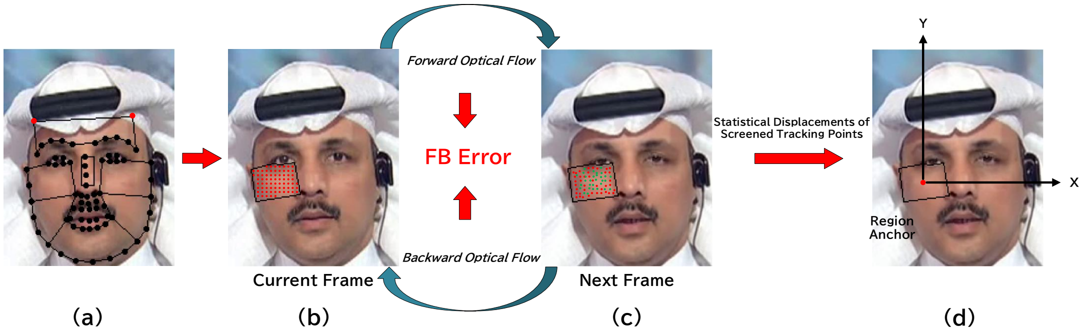

Figure 3. Overview of proposed virtual-anchor-based facial region displacement trajectory extraction method. (a) For each frame, landmarks are first detected and seven ROIs are defined; (b) Tracking points are marked at uniform intervals in each ROI; (c) forward and backward optical flows between current and next frame for tracking points are calculated to obtain forward-backward (FB) errors; (d) points with excellent tracking characteristics (green points) are screened by FB error and statistical displacements of these points are used to represent displacements of virtual anchors in ROIs

4. Trajectory Extraction Method

Existing methods for extracting facial position variation information are not robust. For example, landmark-based methods track the integer coordinates of specific facial landmarks frame-by-frame, which may introduce systematic errors into the sequence since the displacement between frames is usually less than one pixel. Methods based on the Lucas-Kanade algorithm select corner points with good tracking characteristics in the first frame and track the optical flow [23] of these points across the video stream, which may cause optical flow disappearance and tracking point drift since the tracking characteristics change with the movement, and the tracking error accumulates.

To make extraction more robust, we propose using virtual anchors to represent the displacement of facial regions. This will eliminate the effects of singular values and noise on tracking a single point. We also propose screening feature points with excellent tracking characteristics frame-by-frame to prevent error accumulation and optical flow disappearance. An overview of our proposed virtual-anchor-based facial region displacement trajectory extraction method is shown in Figure 3.

Specifically, for each frame, we first define seven regions of interest (ROIs) in accordance with the landmark coordinates extracted using Dlib [19]. Since there is no landmark in the forehead area, in order to cope with different face sizes and head poses, we adopt the method in [26] to adaptively calculate two feature points for use in dividing the forehead ROI. We then mark tracking points t in each ROI at a uniform interval, which is determined by the width w and height h of the detected face rectangle. The intervals in the X and Y directions are $w/40$ and $h/40$, respectively. Next, for each tracking point t_i, we calculate the forward optical flow to obtain position $t_i^{'}$ of the tracking point in the next frame and then calculate the backward optical flow to obtain its tracking offset point $t_i^{''}$. Only the tracking points that are successfully tracked in both the forward and backward optical flows are retained.

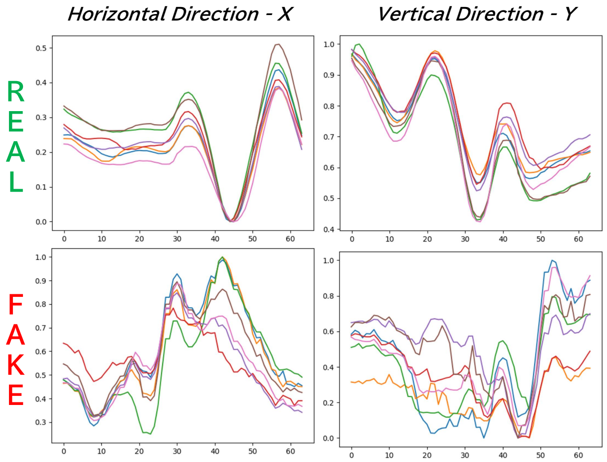

Figure 4. Example real trajectory (upper row) and example fake trajectory (lower row).

The FB error for t_i,

$$FB\_Error_i = |t_i - t_i^{''}| \tag{1}$$

is defined as the position drift of the tracking point after forward and backward optical flow tracking and shows the tracking characteristics of the point. As explained in section 6.4), we discard 50% of the points in each ROI (those with the largest FB errors) and use the statistical displacement (mean and median) values of the remaining points to represent the displacement of the virtual region anchor.

For each video, we record a total of 28 features (mean and median values in x and y directions for 7 ROIs). Due to the inconsistent video lengths and face sizes in datasets, we normalize the trajectories and take 64 frames as the

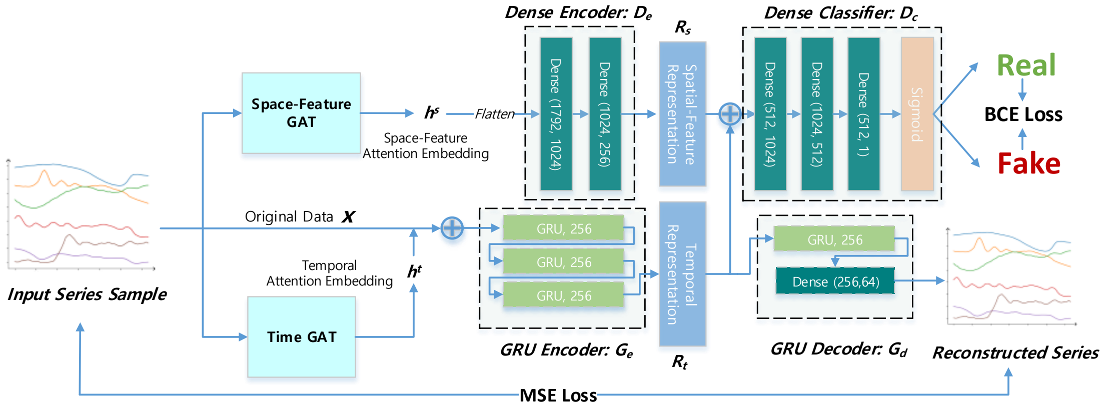

Figure 5. Architecture of proposed FTDN. Model aggregates dynamic graph attention weights using time and space-feature dimensions of trajectory series and obtains corresponding embeddings. GRU and dense encoders are then used to obtain spatial-temporal representations of data, which are subsequently concatenated and used for deepfakes classification. We also use temporal representation to reconstruct input series to ensure accuracy of GRU encoder in sequence feature learning.

window length and 1 s as the sliding length to divide the video trajectory into several sample series with fixed sizes of 28 × 64. As shown in Figure 4, real and fake trajectories can be easily distinguished by the naked eye. For convenience of viewing, we separate the x and y directions and show only the anchor trajectories as calculated using the mean values. Compared with the fake trajectories, the real ones are smoother, and the regional displacements are relatively consistent among ROIs. Moreover, the relative positions between different ROIs for the real trajectory series are basically consistent.

5. Fake Trajectory Detection Network

Intuitively, the facial region trajectory samples can be regarded as an interdependent multivariate time series, where the spatial regions and displacement directions are the variables of the series. Therefore, we can transform the problem of deepfakes video detection into how to mine the spatial-temporal features from the multi-variable time series to distinguish inconsistent patterns in the real and fake samples. To effectively utilize the trajectory series samples to identify the artifacts in forged faces, we use our proposed FTDN. The network architecture is shown in Figure 5.

Specifically, for input series $\boldsymbol{x}$ with size 28 × 64, we first calculate the dynamic graph attention embeddings in the time and space-feature dimensions so that the network can adaptively focus on the correlation between each time step in the trajectory series as well as capture the interaction of spatial information between different facial regions and the coherence in different displacement directions. When calculating the space-feature graph attention embedding, we regard the input series as a set of space-feature sequences $S = \{S_1, S_2, \dots, S_{28}\}, S_i \in \mathbb{R}^{64}$ and take each element S_i in the set as the node vector of the fully connected graph to calculate updated output $\boldsymbol{h}_i^s$:

$$e_{ij}^s = \boldsymbol{a}_s^T LeakyReLU(\boldsymbol{W}_s \boldsymbol{S}_i \oplus \boldsymbol{W}_s \boldsymbol{S}_j)$$

$$\alpha_{ij}^s = Softmax_j(e_{ij}^s)$$

$$\boldsymbol{h}_i^s = \sigma(\sum_j \alpha_{ij}^s \boldsymbol{W}_s \boldsymbol{S}_j) \tag{2}$$

where σ is a sigmoid function, and symbol "$\oplus$" indicates the concatenation of vectors. For space-feature node vectors, we set learnable linear transform matrix $\boldsymbol{W}_s \in \mathbb{R}^{64\times64}$ and attention vector $\boldsymbol{a}_s^T \in \mathbb{R}^{128}$ so that the updated output $\boldsymbol{h}_i^s \in \mathbb{R}^{64}$ after aggregation. This enables us to obtain the space-feature embedding $\boldsymbol{h}^s \in \mathbb{R}^{28\times64}$, consistent with the original data. Similarly, when calculating the time graph attention embedding, we regard the features of each time step in the trajectory series as a node vector of the fully connected graph and get a set of sequences $T = \{T_1, T_2, \dots, T_{64}\}, T_i \in \mathbb{R}^{28}$. To ensure that the output time graph attention embedding $\boldsymbol{h}^t$ has the same size as the original data, we set learnable linear transform matrix $\boldsymbol{W}_t \in \mathbb{R}^{28\times28}$ and attention vector $\boldsymbol{a}_t^T \in \mathbb{R}^{56}$.

Next, we concatenate original series $\boldsymbol{x}$ and time graph attention embedding $\boldsymbol{h}^t$ to form a series group with a size of 56 × 64 and send it to a GRU encoder. After capturing and modeling the correlation between each independent time step, the encoder outputs a 256-dimensional time descriptor $\boldsymbol{R}_t$ for temporal feature representation. For space-feature attention embedding $\boldsymbol{h}^s$, we flatten the data and send it to a dense encoder to obtain the 256-dimensional spatial-feature representation $\boldsymbol{R}_s$. Process can be formulated as

$$\boldsymbol{R}_s = D_e(flatten(\boldsymbol{h}^s))$$

$$\boldsymbol{R}_t = G_e(\boldsymbol{x} \oplus \boldsymbol{h}^t) \tag{3}$$

in which D_e and G_e represent the dense and GRU encoders, respectively. We do not send $\boldsymbol{h}^s$ with $\boldsymbol{h}^s$ and $\boldsymbol{x}$ to the GRU encoder together since each node vector in the space-feature set does not have invariance in the time dimension. During graph information aggregation, the time independence of the reconstructed output is removed, which creates unnecessary redundancies for temporal representation learning.

Finally, we concatenate the two potential representations and input them into the fully connected layers and sigmoid activation. We use binary cross entropy (BCE) as classification loss for real-fake binary classification:

$$L_{BCE} = BCE(D_c(\boldsymbol{R}_s \oplus \boldsymbol{R}_t), \hat{y}) \tag{4}$$

where D_c is the dense classifier, and $\hat{y} = \{0, 1\}$ is the attribute label of the input series sample. Furthermore, to enhance the modeling of the time descriptor by the GRU encoder, we added an auxiliary learning task. The temporal representation encoded by the GRU encoder is sent to a GRU decoder to reconstruct the multi-variable time series, and then the mean square error (MSE) is used as the reconstruction loss to restrict the similarity between the reconstructed sequence and the input sequence $\boldsymbol{x}$ so as to improve the accuracy of learning the sequence features by the GRU module:

$$L_{MSE} = MSE(G_d(\boldsymbol{R}_t), \boldsymbol{x}) \tag{5}$$

where G_d is a GRU decoder composed of a GRU layer and a fully connected layer. To sum up, the total loss consists of two parts:

$$L = L_{BCE} + L_{MSE} \tag{6}$$

6. Evaluation

6.1. Dataset settings

To evaluate our proposed FTDN, we mainly used the FaceForensics++ (FF++) dataset [27] for training and testing. The FF++ dataset consists of a source video sub-dataset extracted from Youtube and several forged video sub-datasets generated by different manipulation techniques, each sub-dataset contains 1000 manipulated videos. For each video in the dataset, the publisher had given three compression versions: uncompressed (raw), moderately-compressed (c23), and strongly-compressed (c40).

We combined the source videos in FF++ dataset with each of the four kinds of manipulated videos (Deepfakes, FaceSwap, Face2Face and NeuralTextures) to form four sub-datasets for our binary classification tasks. The sub-datasets are referred to as DF, FS, F2F, and NT, respectively. For each sub-dataset, we created a training set, a validation set, and a test set at a ratio of 8:1:1. Since the FaceSwap and NeuralTextures videos were slightly shorter than the real videos, when constructing the FS and NT sub-datasets, we used only the top 80% of the real videos to ensure balance between positive and negative samples. In addition, we separately selected 100 videos with the same index as the DF test set from the FaceShifter to form an unseen sub-dataset called FSH, which is used to test the cross data domain generalization ability of our method. When generating the trajectory samples, we discarded a frame if the frame was not captured, a face was not detected, or the number of points with successful forward and backward optical flow tracking was less than 20% of the total number of tracking points. If more than ten frames in a video were discarded, we would separate the clips that can continuously detect the face and use them in the test phase.

Baseline	**Sub-Datasets**				
	DF	**FS**	**F2F**	**NT**	**FSH**
FakeCatcher [9]	92.5%	94.5%	93.5%	79.5%	63.0%
PPG Cell [10]	94.5%	93.0%	94.5%	77.0%	70.0%
RCN [28]	97.0%	95.5%	95.5%	85.5%	65.5%
MesoNet [3]	97.0%	94.0%	92.0%	83.5%	72.0%
Xception [8]	99.0%	97.0%	97.5%	**93.0%**	65.5%
Capsule-F [25]	96.5%	97.5%	96.5%	89.0%	68.5%
LRNet [31]	98.5%	98.0%	92.0%	86.5%	70.0%
FTDN(Sample)	99.32%	99.16%	98.14%	90.66%	75.13%
FTDN(Video)	**99.5%**	**99.5%**	**98.5%**	92.5%	**75.0%**

Table 1. The binary classification accuracy comparison on each sub-dataset, and the generalization comparison on unseen dataset.

6.2. Accuracy Comparison with Baseline

We evaluated the accuracy of the proposed FTDN by using seven state-of-the-art deepfakes detection methods as our baselines: FakeCatcher [9], PPG Cell [10], recurrent convolutional network (RCN) proposed by Sabir *et al.* [28], MesoNet [3], Xception [8], Capsule-Forensics [25] and LRNet [31]. For Xception, the benchmark for the FF++ dataset, we modified the final classification layer of the pre-trained model and fine-tuned the parameters by using the FF++ dataset following the guidance of Rossler *et al.* For MesoNet, Capsule-Forensics and LRNet, we used the source code provided by the authors to retrain the model and validate the performance by our sub-datasets. For the other methods, we reproduced their works according to the paper. Xception, MesoNet, and Capsule-Forensics were designed for image-level detection, so we had to made some adjustments to adapt them to our video task. Specifically, for a test video, we extracted an image every ten frames to form a sample set, used the well-trained model to predict all the images in the set, and determined the attributes of the test video by majority voting. Similarly, for our method, since each video was divided into several series samples, we used FTDN in the test phase to predict all series samples and determined the result by majority voting.

The binary classification accuracies of our proposed FTDN on each sub-dataset are listed in Table 1, including the sample-level and video-level accuracies. The best result for each sub-dataset is shown in bold. FTDN at the video level had the highest accuracy on the DF, FS, and F2F sub-datasets (99.5%, 99.5%, and 98.5%, respectively) and had accuracy close to that of the FF++ benchmark (Xception), which had the highest accuracy (93.0%), on the NT sub-dataset. Compared with FakeCatcher and PPG Cell, which use facial color series abnormalities to detect deepfakes, the facial trajectory series we extracted is more specific and has better detection performance. Furthermore, in contrast to the LRNet model, which also exposes deepfakes by the geometric information, the GAT mechanism in our method effectively highlights abnormal fluctuations in temporal and spatial dimensions, making our method have better detection performance on F2F and NT sub-datasets.

To verify the generalization ability of our method, which is still a big challenge for all detectors, we trained FTDN and other baselines with DF training set, and check the performance of these models on unseen FSH sub-dataset. As shown in the last column of Table 1, our method achieved the best detection accuracy of 75%. In contrast, FakeCatcher showed the worst performance (i.e. 63%). Although Xception was better than other baselines in the binary classification task, its generalization ability was not as good as MesoNet (i.e. 72%) with lightweight parameters. RCN model achieved 65.5% accuracy in cross data domain detection, slightly lower than Capsule-Forensics (i.e. 68.5%),PPG Cell and LRNet(i.e. 70%).

6.3. Robustness against video compression

Our method models the spatial-temporal features of the displacement trajectory of different facial areas and capture the errors introduced by affine transformation during image blending, which will not be affected by video compression theoretically. However, low image quality has two effects: 1)inaccurate landmark location, which affects the division of ROIs and the feature points involved in tracking; 2)bad video conditions, which add extra noise to the optical flow calculation, which affects the accuracy of feature point tracking. Specific regions of the displacement trajectory under different compression levels are plotted in Figure 6 for the same Deepfakes video. Increasing the degree of compression produced errors in the landmark locations, resulting in a spatial offset in the frame-by-frame ROI division, which changed the relative position of the trajectory. Furthermore, the fuzziness of the pixel information affected the feature point tracking accuracy, which made it difficult to capture the subtle spatial artifacts in local facial areas that resulted from manipulation. This is evidenced by the ”smoothing” of the high-frequency burr noise in the areas enclosed by the red box. As noted above, these high-frequency burrs are attributed to the spatial error caused by the forced inverse affine transformation, and are further manifested as the rapid changes in motion trend of local regions during frame-by-frame manipulation, which might provide important evidence for model discrimination.

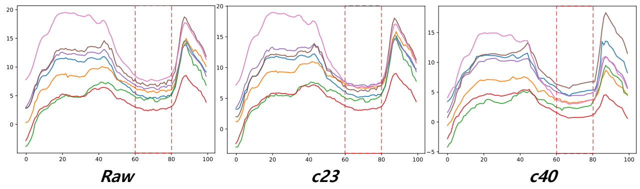

Figure 6. DeepFake trajectory under different compression levels.

Baseline	DF		F2F	
	c23	c40	c23	c40
PPG cell [10]	89.43%	79.50%	86.41%	77.84%
MesoNet [3]	93.62%	90.19%	91.98%	80.57%
Xception [8]	96.55%	**92.17%**	95.21%	87.39%
Capsule-F [25]	94.98%	91.06%	94.82%	**89.10%**
FTDN(ours)	**98.05%**	83.55%	**96.32%**	83.19%

Table 2. Accuracy comparison on different compression levels.

We tested the anti-compression robustness of our method by using videos in the DF and F2F sub-datasets with different compression levels. The results in Table 2 show that our method maintained excellent performance for moderately-compressed (c23) videos and achieved the highest detection accuracy compared with the baselines. However, for strongly-compressed (c40) videos, its accuracy was severely degraded, and the detection accuracy was lower than that of some image-based methods.

6.4. Ablation Study

As mentioned above, when extracting the facial displacement trajectory, we discard 50% of the tracking points (those with the largest FB errors), and use the mean and median values of the displacement of the remaining tracking points in the region to represent the displacement of the virtual anchor. We investigated the effectiveness without discard, the effectiveness of using the mean value, and the effectiveness of using the median value through an ablation experiment. As shown in Table 3 discarding 50% of the tracking points with poor tracking characteristics effectively improved detection accuracy. Basically, the effect of using a trajectory based on the mean was better than that of using one based on the median, and a combination of them achieved the best result.

To determine the effectiveness of each component in our network architecture, we conducted ablation experiments on our FTDN, focusing on the effectiveness of the Time GAT (T), the Space-Feature-GAT (SF), and the GRU encoder. The results in Table 4 show that the model relies on the GRU encoder to capture the temporal features of the trajectory series. The Time-GAT and Space-Feature-GAT

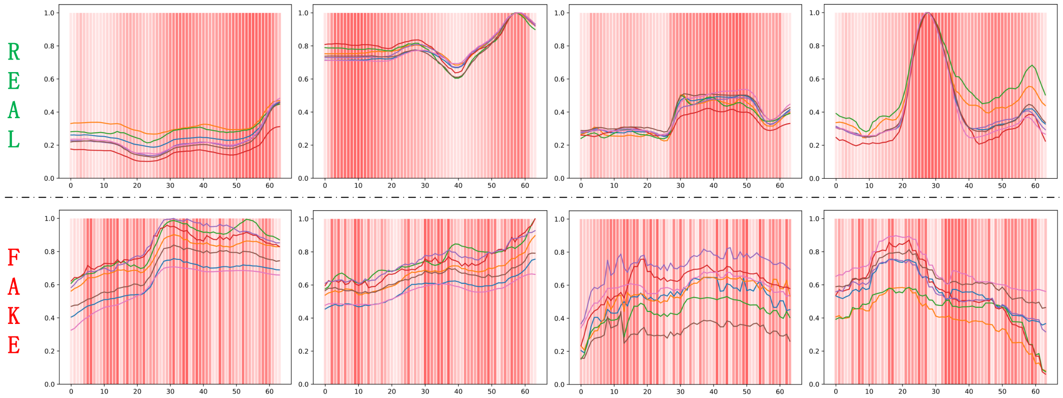

Figure 7. We use the gradient visualization of real and fake samples to highlight the time steps in the series that contribute significantly to the final classification. The darker the red, the greater impact of the time step on model discrimination.

Sub-Dataset	w/o discard	mean+median	mean	median
DF	w	**99.32%**	98.70%	98.28%
	o	99.03%	99.15%	97.97%
FS	w	**99.16%**	99.07%	98.80%
	o	97.11%	97.11%	95.89%
F2F	w	**98.14%**	96.28%	96.14%
	o	96.80%	95.01%	96.17%
NT	w	**90.66%**	83.34%	85.65%
	o	87.37%	84.03%	86.72%

Table 3. Results of ablation study on facial region displacement trajectory extraction method.

Settings	**Sub-Dataset**			
	DF	**FS**	**F2F**	**NT**
FTDN	99.32%	99.16%	98.14%	90.66%
w/o T	98.68%	97.65%	93.55%	84.81%
w/o SF	98.90%	97.50%	96.21%	88.79%
w/o T and SF	98.72%	97.40%	93.55%	82.11%
w/o GRU	52.80%	49.32%	56.77%	51.12%

Table 4. Results of ablation study on FTDN model.

weight the importance of the sequence information, which helps the GRU improve detection accuracy. Moreover, the contribution of graph attention to F2F and NT is much greater than that to DF and FS. We attribute this to Deepfakes and FaceSwap having a larger area for face swapping, which increases the deviations in their trajectories, enabling the GRU to more easily capture the pattern differences between real and fake samples. In contrast, Face2Face and NeuralTextures affect only subtle areas related to facial expressions, so the effect on the trajectory is relatively minor. The artificial spatial-temporal attention mechanism can thus effectively improve the accuracy of the model.

6.5. Visualization

Inspired by the concept of gradient-weighted class activation mapping [30], we used gradient information to visualize the focus of the model on samples. Figure 7 shows the contribution of trajectory information for each time step to model discrimination (the darker the color, the greater the contribution). The model identifies real samples by identifying continuous, smooth, and stable trajectory patterns, so the time steps focused on for real samples are often successive. In contrast, the model identifies fake samples by identifying the burr noise caused by rapid changes in motion trend and focuses on the time steps in which the relative positions of different ROI sequences change. Therefore, for the fake trajectory series, the region focused on by the model is relatively discrete.

7. Conclusion

We have devised a deepfakes detection method based on the facial region displacement trajectory. Specifically, we propose using virtual-anchor-based region displacement trajectory extraction to obtain the spatial-temporal representation of different facial areas robustly and accurately. We have also constructed a fake trajectory detection network based on dual-stream spatial-temporal graph attention and a gated recurrent unit backbone that converts the deepfakes detection task into a binary classification problem for a multi-variable time series. Our detection method exhibited competitive performance on samples from the FaceForensics++ dataset.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grants JP16H06302, JP18H04120, JP20K23355, JP21H04907, and JP21K18023, and by JST CREST Grants JPMJCR18A6 and JPMJCR20D3, including the AIP challenge program, Japan.

References

[1] FaceApp. https://faceappdownload.org, 2017.

[2] ZAO. https://apps.apple.com/cn/app/id1465199127, 2019.

[3] Darius Afchar, Vincent Nozick, Junichi Yamagishi, and Isao Echizen. Mesonet: a compact facial video forgery detection network. In *2018 IEEE international workshop on information forensics and security (WIFS)*, pages 1–7. IEEE, 2018.

[4] Shruti Agarwal, Hany Farid, Yuming Gu, Mingming He, Koki Nagano, and Hao Li. Protecting world leaders against deep fakes. In *CVPR workshops*, volume 1, page 38, 2019.

[5] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

[6] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[8] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[9] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Fakecatcher: Detection of synthetic portrait videos using biological signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[10] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. How do the hearts of deep fakes beat? deep fake source detection via interpreting residuals with biological signals. In *2020 IEEE international joint conference on biometrics (IJCB)*, pages 1–10. IEEE, 2020.

[11] Naser Damer, Viola Boller, Yaza Wainakh, Fadi Boutros, Philipp Terhörst, Andreas Braun, and Arjan Kuijper. Detecting face morphing attacks by analyzing the directed distances of facial landmarks shifts. In *German Conference on Pattern Recognition*, pages 518–534. Springer, 2018.

[12] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4027–4035, 2021.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[14] David Güera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2018.

[15] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] Fumitada Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on acoustics, speech, and signal processing*, 23(1):67–72, 1975.

[17] Fazle Karim, Somshubra Majumdar, and Houshang Darabi. Insights into lstm fully convolutional networks for time series classification. *IEEE Access*, 7:67718–67725, 2019.

[18] Rohit J Kate. Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery*, 30(2):283–312, 2016.

[19] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014.

[20] Marissa Koopman, Andrea Macarulla Rodriguez, and Zeno Geradts. Detection of deepfake video manipulation. In *The 20th Irish machine vision and image processing conference (IMVIP)*, pages 133–136, 2018.

[21] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[22] Yuezun Li and Siwei Lyu. Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*, 2018.

[23] Bruce D Lucas, Takeo Kanade, et al. *An iterative image registration technique with an application to stereo vision*, volume 81. Vancouver, 1981.

[24] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Timenet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*, 2017.

[25] Huy H Nguyen, Junichi Yamagishi, and Isao Echizen. Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2307–2311. IEEE, 2019.

[26] Xuesong Niu, Zitong Yu, Hu Han, Xiaobai Li, Shiguang Shan, and Guoying Zhao. Video-based remote physiological measurement via cross-verified feature disentangling. In *European Conference on Computer Vision*, pages 295–310. Springer, 2020.

[27] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

[28] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi, and Prem Natarajan. Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3(1):80–87, 2019.

[29] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[30] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[31] Zekun Sun, Yujie Han, Zeyu Hua, Na Ruan, and Weijia Jia. Improving the efficiency and robustness of deepfakes detection through precise geometric features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3609–3618, 2021.

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[33] Xin Yang, Yuezun Li, and Siwei Lyu. Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8261–8265. IEEE, 2019.

[34] Xin Yang, Yuezun Li, Honggang Qi, and Siwei Lyu. Exposing gan-synthesized faces using landmark locations. In *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pages 113–118, 2019.

[35] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 841–850. IEEE, 2020.