



# Deepfake Video Detection Based on Improved CapsNet and Temporal–Spatial Features

Tianliang Lu\*, Yuxuan Bao and Lanting Li

People's Public Security University of China, Beijing, 100038, China  
\*Corresponding Author: Tianliang Lu. Email: lutianliang@ppsuc.edu.cn  
Received: 02 August 2022; Accepted: 22 November 2022

**Abstract:** Rapid development of deepfake technology led to the spread of forged audios and videos across network platforms, presenting risks for numerous countries, societies, and individuals, and posing a serious threat to cyberspace security. To address the problem of insufficient extraction of spatial features and the fact that temporal features are not considered in the deepfake video detection, we propose a detection method based on improved CapsNet and temporal–spatial features (iCapsNet–TSF). First, the dynamic routing algorithm of CapsNet is improved using weight initialization and updating. Then, the optical flow algorithm is used to extract interframe temporal features of the videos to form a dataset of temporal–spatial features. Finally, the iCapsNet model is employed to fully learn the temporal–spatial features of facial videos, and the results are fused. Experimental results show that the detection accuracy of iCapsNet–TSF reaches 94.07%, 98.83%, and 98.50% on the Celeb-DF, FaceSwap, and Deepfakes datasets, respectively, displaying a better performance than most existing mainstream algorithms. The iCapsNet–TSF method combines the capsule network and the optical flow algorithm, providing a novel strategy for the deepfake detection, which is of great significance to the prevention of deepfake attacks and the preservation of cyberspace security.

**Keywords:** Deepfake detection; CapsNet; optical flow algorithm; temporal–spatial features

## 1 Introduction

In recent years, the optimization of deep neural networks and the improved performance of the graphics processing unit (GPU) led to the rapid development of deepfake technology, allowing the automatic creation of more realistic forged videos. Using deepfake technology, one can superimpose the facial image of the target person onto the corresponding position of the face of the original person in the video, thus creating a video of the target person making statements or performing actions that did not actually take place, in order to confuse viewers [1].



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deepfake videos have spread rapidly with the help of online platforms, posing risks to individuals, societies, and countries. In June 2019, DeepNude software used the deepfake technology to generate nude pictures of female victims, violating the privacy and reputation of more than 104,000 women. Telecommunication fraud, performed through forged videos and multiple rounds of dialogues, has likewise become a trend [2]. Social engineering takes advantage of the victims' psychological weaknesses to carry out the deception. If combined with deepfake technology, social engineering may achieve a better deception effect [3]. For example, in August 2019, criminals created a voice model of a German energy company's president and successfully defrauded the company's UK branch, stealing EUR 220,000. During the U.S. election in April 2020, former U.S. President Donald Trump retweeted a video made by deepfake technology to vilify current President Joe Biden. Thus, politicians can use the technology to disrupt elections, manipulate public opinion and trigger a crisis of confidence [4,5]. Therefore, the timely detection of deepfake videos is among the key tasks in cyberspace security.

Mainstream deepfake video detection methods are divided into two categories: one based on video intraframe feature differences that learns the spatial features of a single frame and makes a comprehensive decision based on the prediction results of each frame, and the other based on video interframe feature differences, which treats deepfake videos as time-series data, uses the time-series algorithm to capture the differences of the temporal features between real and fake videos frame-by-frame, and then classifies them.

In this study, we propose iCapsNet-TSF, a deepfake video detection method based on improved CapsNet and the fusion of temporal-spatial features. The iCapsNet-TSF method combines the capsule network and optical flow algorithm to detect deepfake videos for the first time. Compared with previous related studies, our method makes comprehensive decisions based on the temporal and spatial features of facial images, which significantly improves the detection accuracy.

The organization of our paper is as follows. Section 1 introduces the basic concepts and relevant background of deepfake technology. Section 2 introduces the studies related to the deepfake video detection method. Section 3 introduces the detection method proposed in this study. Section 4 presents the experiments used to evaluate the effectiveness of the proposed method through a variety of indexes. Section 5 lists the conclusions, summarizing the work conducted in this study and considering future directions of this field.

## 2 Related Work

Deepfake video detection methods based on feature extraction can be divided into detection based on video intra- and interframe features. The procedures for both are as follows: (1) a key frame extraction and face detection algorithm are used to convert video data into facial image data, and then (2) a deep neural network is used to extract the spatial or temporal feature vector of the facial image, and, finally, (3) a classification function is used to distinguish between real and fake faces.

The detection method based on the feature extraction uses the deep learning technology, which involves the learning of massive datasets and the use of a back-propagation algorithm to identify the optimal detection model, significantly improving the detection by saving manpower and more accurately distinguishing falsified faces.

### 2.1 Detection Based on Video Intraframe Features

The detection method based on video intraframe features refers to the use of traditional algorithms, machine learning algorithms, or deep learning models to learn the spatial features of a single

frame of a deepfake video and then making a comprehensive decision based on the prediction result of each frame. This method can fully extract spatial features, whereas it cannot fully use the temporal features of deepfake videos, because it ignores the fact that the videos themselves rely on temporal data. The essence of this method is the image detection, which can detect forged videos with high accuracy while minimizing parameter redundancy.

In addition to other data, traditional algorithms consider the attributes of the video, such as the frame rate and brightness. Koopman et al. [6] calculated normalized cross-correlation scores to distinguish the deepfake videos based on the differences in the light perception noise during the camera shooting. Luo et al. [7] found that the high-frequency signal of the image removed the color texture, which helps to more effectively distinguish real and fake videos. Therefore, high-frequency signal features were fused with original RGB features by the attention mechanism, and the videos were finally classified.

Machine learning algorithms are often used to classify falsified faces by extracting a one-dimensional vector that represents facial features. On the basis of large differences in the colors of eyes, shadows appearing on the edges of the nose, and littery geometric rules of teeth on falsified faces, Matern et al. [8] extracted facial feature vectors via computer vision methods, such as color histograms and color aggregation vectors, and then used the k-nearest neighbor (KNN) algorithm to classify them. Yang et al. [9] found that head poses of the original face change during face tampering, and the authors used the differences between the feature vectors of the tampered parts and the whole face as a criterion to obtain a classification with the help of the support vector machine (SVM) algorithm. Durall et al. [10] extracted the two-dimensional power spectrum of video frames via the discrete Fourier transform, which is compressed into a one-dimensional feature vector using orientation averaging, and then classified using a logistic regression algorithm.

When detecting deepfake videos, traditional machine learning algorithms must often extract facial features manually. Furthermore, because of classifier limitations, it is difficult to extract deep pixel-level spatial features of the image.

Therefore, generative networks and convolutional neural networks (CNNs) are gradually used to detect deepfake videos. Cozzolino et al. [11] proposed a novel approach that learns temporal facial features by means of metric learning coupled with an adversarial training strategy. Using a three-dimensional morphable model (3DMM), the authors processed videos of different identities on a frame-by-frame basis and trained the Temporal ID Network to embed the extracted features. To incentivize this network to focus on temporal aspects rather than visual cues, they jointly trained the 3DMM Generative Network to transform extracted features to fool its discriminative counterpart.

Afchar et al. [12] argued that the images' low-layer noise features degrade with video compression, and high-layer semantic features are difficult to use as the basis for detection. Therefore, they proposed the mesoscopic network (MesoNet) combined with the Inception module to classify deepfake videos. Zhou et al. [13] proposed a two-stream CNN network employing RGB and noise convolution layers to extract pixel and noise features of video frames, and then fuse the two features to improve the detection accuracy. Nguyen et al. [14] used CapsNet to learn detailed information about the facial pose (position, hue, texture) and extract richer spatial features of the faces. Zhu et al. [15] proposed a combination of direct light and common texture to detect deepfake videos. At the same time, the supervised attention mechanism highlights the tampered areas to detect facial details. Wang et al. [16] proposed an attention-based data enhancement framework to guide detectors to refine and expand their attention. This method tracks and occludes the top-N sensitive areas of the faces and encourages the detectors to further explore previously ignored areas to obtain more accurate results.

With the development of the deep learning model, relevant studies presented at the top conferences in the last two years include the following. Li et al. [17] proposed the Frequency-aware Discriminative Features Learning (FDFL) framework, which uses a new single-center loss to reduce the intraclass changes in natural faces, so as to increase interclass differences in the embedded space. Simultaneously, they developed an adaptive frequency feature generation module, which mines subtle artifacts from the frequency domain in a data-driven manner. Zhao et al. [18] designed a Patch-wise Consistency Learning (PCL) branch to send the middle layer of the backbone network to different encoders and subsequently dot-multiply the results to provide additional supervision information for the backbone network and guide the model to pay attention to the similarity between forged and adjacent areas. Dong et al. [19] proposed an Identity Consistency Transformer (ICT), a novel facial video forgery detection method focusing on high-level semantics—specifically, identity information—and detecting a suspicious face by determining the identity inconsistency in inner and outer facial regions. The ICT incorporates a consistency loss to determine the identity consistency and exhibits superior generalization ability across different deepfake datasets.

Shao et al. [20] proposed the Seq-DeepFake Transformer (SeqFakeFormer) to detect forged images. First, they captured spatial manipulation traces of the image through self-attention modules in the transformer encoder, and then added the Spatially Enhanced Cross-Attention (SECA) module to generate different spatial weight maps for corresponding manipulations to carry out cross-attention. Wang et al. [21] proposed a novel Shuffled Style Assembly Network (SSAN) to extract and reassemble different contents and style features for a stylized feature space of facial images. Then, to obtain a generalized representation, a contrastive learning strategy was developed to emphasize liveness-related style information while suppressing domain-specific information. Gu et al. [22] proposed a Progressive Enhancement Learning Framework (PELF), which utilized both RGB information and fine-grained frequency information. Sun et al. [23] proposed Dual Contrastive Learning (DCL), which specially constructed positive and negative paired data and performed designed contrastive learning at different granularities to learn generalized feature representation. Specifically, they proposed Inter-Instance Contrastive Learning (Inter-ICL) based on the hard sample selection strategy to promote task-related discriminant feature learning.

## 2.2 Detection Based on Video Interframe Features

Because deepfake videos are synthesized frame-by-frame during the generation process, it is difficult to consider the previously forged frame sequence. People in the forged video will have a significantly lower frequency of blinking, and facial movements will be uncoordinated. Besides, the face brightness will change when the video is played frame-by-frame. Therefore, it can be captured by the algorithms considering temporal features.

The detection method based on video interframe features usually classifies deepfake videos according to the temporal features. However, it is sensitive to the length of frames, cannot effectively extract temporal features from the videos with short playback times, and lacks the learning of detailed spatial features of forged faces.

Sabir et al. [24] adopted the recurrent convolutional strategy. To detect deepfake videos, the CNN was used to extract the facial spatial features of each frame of the video, after which the recurrent neural network (RNN) was used to learn the time-series changes about facial features. Several studies found that the blinking frequencies of real and fake faces differ. Li et al. [25] used this observation for the identification, extracting the eye-distinguishing features of faces using the visual geometry group (VGG) network, and then used long short-term memory (LSTM) to learn the blinking frequency



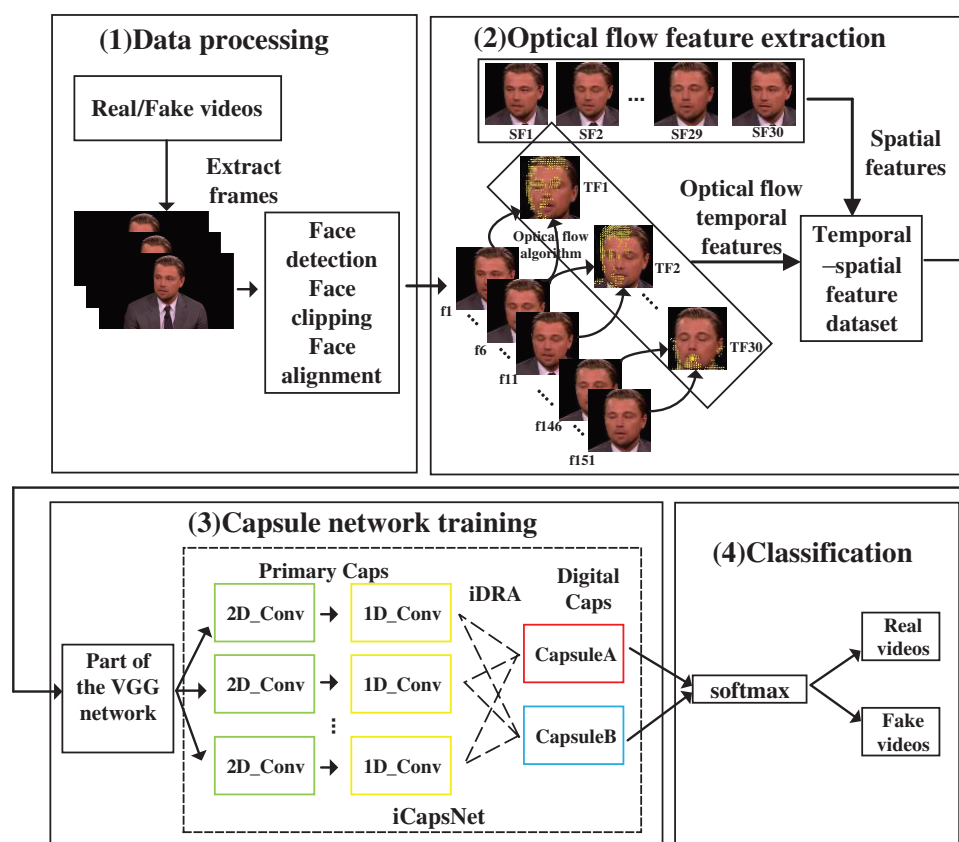
features of real and fake faces. Amerini et al. [26] used PWC-Net and LK algorithms to extract the optical flow vectors, and VGG16 was used to capture the difference between real and fake faces. For the preprocessing, Sun et al. [27] designed a calibration module to obtain a more accurate sequence of facial landmarks. They embedded the landmarks into two types of feature vectors, and then used the two-stream RNN to mine the time information and determine its authenticity. One feature vector serves to simulate the movement pattern of the facial shape, and the other is regarded as the velocity pattern used to capture the discontinuity of time. The high-level temporal semantic features of lips are difficult to forge by existing methods. Haliassos et al. [28] first pretrained the ResNet-18 network and the time-series network in the lip-reading task, fixed the ResNet-18 network, and trained only the time-series network so as to extract lip features that can help determine real and fake videos and realize the most advanced generalization of the forgery type. Gu et al. [29] studied the local motion information in videos and proposed a new video sampling unit, “Snippet”, which contains some local continuous video frames. Furthermore, the Intra-Snippet Inconsistency Module (Intra-SIM) and Inter-Snippet Interaction Module (Inter-SIM) were carefully designed to establish the inconsistent dynamic modeling framework. Specifically, Intra-SIM uses two-way time difference operations and learnable convolution kernels to mine the subtle motion in each “Snippet”. Inter-SIM is then used to facilitate the information interaction among “Snippets” to form a global representation. Table 1 shows the recent deepfake detection algorithms presented at the top conferences in the last two years.

**Table 1:** Recent deepfake detection algorithms

Algorithm	Published year	Published conference	Accuracy (%)
Time-series ID network and the 3DMM generative network [11]	2021	ICCV	90.03–98.76
Frequency-aware Discriminative Features Learning (FDFL) framework [17]	2021	CVPR	89.00–99.43
Patch-wise Consistency Learning (PCL) [18]	2021	ICCV	96.45–98.05
Identity Consistency Transformer (ICT) [19]	2022	CVPR	93.17–99.25
Seq-DeepFake Transformer (SeqFakeFormer) and Spatially Enhanced Cross-Attention (SECA) module [20]	2022	ECCV	94.23–98.78
Shuffled Style Assembly Network (SSAN) [21]	2022	CVPR	93.63–98.75
Progressive Enhancement Learning Framework (PELF) [22]	2022	AAAI	90.52–97.63
Dual Contrastive Learning (DCL) [23]	2022	AAAI	91.66–98.97
Two-stream RNN [27]	2021	CVPR	95.70–99.90
ResNet-18 network and the time-series network using temporal features of lips [28]	2021	CVPR	92.50–98.10
Intra-Snippet Inconsistency Module (Intra-SIM) and Inter-Snippet Interaction Module (Inter-SIM) [29]	2022	AAAI	94.28–99.28

### 3 Deepfake Video Detection Method

In this study, a deepfake video detection method, iCapsNet–TSF, is proposed, as shown in Fig. 1.



**Figure 1:** Deepfake video detection based on improved CapsNet and temporal-spatial features

The deepfake video detection method based on improved CapsNet and temporal-spatial features includes four stages: data processing, optical flow feature extraction, capsule network training, and classification of real and fake faces.

- (1) Data processing: Key frame extraction and face detection algorithms are used to transform video data into image data to form a spatial feature dataset. A sharpening operation is used to refine the image. Gaussian blur is added to reduce image noises. Data on the image are standardized and normalized to enhance the generalization ability of the model.
- (2) Optical flow feature extraction: Using the Lucas–Kanade (LK) optical flow algorithm, the best frame extraction strategy is selected to extract the temporal features of videos to form a dataset of temporal-spatial features. This dataset provides rich image features for the model, such that we can fully learn the temporal-spatial feature distribution of facial images.
- (3) Capsule network training: The VGG network is used to preliminarily extract facial features. The feature image is sent to the capsule network for training. During the training process, the dynamic routing algorithm (DRA) of the capsule network is improved in terms of the weight initialization and weight updating. Then, the improved dynamic routing algorithm (iDRA) and the corresponding improved CapsNet (iCapsNet) model are proposed.
- (4) Classification: After training the model on the temporal-spatial feature dataset, the cross-entropy loss function is used to evaluate the difference between predicted and true values. Then, the softmax function is used to obtain a binary classification of real and fake videos.

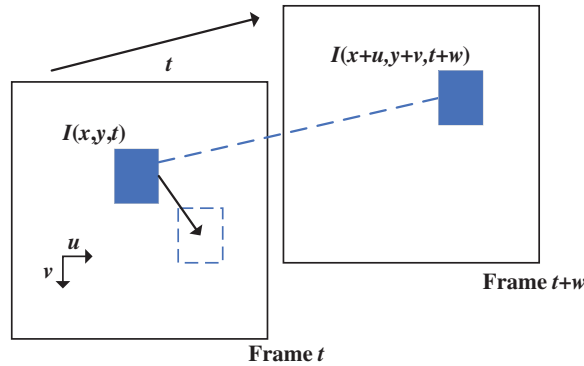
### 3.1 Optical Flow Feature Extraction

#### 3.1.1 Optical Flow Algorithm

Optical flow is defined as the instantaneous velocity of the pixel motion in sequential images [30]. The optical flow algorithm analyzes the correlation between adjacent frames by estimating their optical flow. Therefore, the optical flow algorithm is often used to process video data.

There are two assumptions for calculating the optical flow between adjacent frames. Assumption (1): the brightness and color of the same pixel in two adjacent frames of the video remain stable. Assumption (2): the video is continuous, and the position of the target pixel does not change significantly within a short time.

Let the luminance of the pixel point  $(x, y)$  in the initial frame at moment  $t$  be  $I(x, y, t)$ . The displacement at time  $\delta_t$  is  $(\delta_x, \delta_y)$ , such that the brightness in the adjacent frame is  $I(x + \delta_x, y + \delta_y, t + \delta_t)$ . The displacement of pixels in three dimensions can be identified intuitively, as shown in Fig. 2.



**Figure 2:** Visual presentation of pixel movement

According to Assumption (1), the brightness of the pixel remains constant during the motion of adjacent frames, as shown in Eq. (1).

$$I(x, y, t) = I(x + \delta_x, y + \delta_y, t + \delta_t) \quad (1)$$

Expanding the right-hand side of Eq. (1) using a first-order Taylor series yields Eq. (2):

$$I(x + \delta_x, y + \delta_y, t + \delta_t) = I(x, y, t) + \frac{\partial I(x, y, t)}{\partial x} \delta_x + \frac{\partial I(x, y, t)}{\partial y} \delta_y + \frac{\partial I(x, y, t)}{\partial t} \delta_t + \varepsilon \quad (2)$$

where  $\varepsilon$  represents a second-order and higher-order infinitesimal constant. As it is assumed that the pixel motion displacement is small,  $\varepsilon$  can be ignored. Dividing Eq. (2) by  $\delta_t$ , we obtain Eq. (3).

$$\frac{\partial I(x, y, t)}{\partial x} \frac{\delta_x}{\delta_t} + \frac{\partial I(x, y, t)}{\partial y} \frac{\delta_y}{\delta_t} + \frac{\partial I(x, y, t)}{\partial t} = 0 \quad (3)$$

Let  $V_x = \frac{\delta_x}{\delta_t}$  and  $V_y = \frac{\delta_y}{\delta_t}$  denote the instantaneous velocity of the pixel moving along the  $x$ - and  $y$ -axis, respectively. Let  $I_x = \frac{\partial I}{\partial x}$ ,  $I_y = \frac{\partial I}{\partial y}$ , and  $I_t = \frac{\partial I}{\partial t}$  denote the difference in the brightness of the pixel along the  $x$ -,  $y$ -, and  $t$ -axes, respectively and transpose Eq. (3) to obtain Eq. (4):

$$I_x V_x + I_y V_y = -I_t \quad (4)$$

where  $V_x$  and  $V_y$  are the optical flow to be solved, and  $I_x$ ,  $I_y$  and  $I_t$  can be solved from the image itself. However, Eq. (4) has two unknowns, which could not be solved yet. The LK optical flow algorithm solves this difficult problem. The algorithm adds a new assumption to the previous two assumptions, which is that the displacement of all pixel points is similar in the  $m \times m$  region around the domain of the pixel point to be solved. Eq. (5) is obtained by transferring Eq. (4) into a matrix, where  $n = m^2$ .

$$\begin{bmatrix} I_{x1} & I_{y1} \\ I_{x2} & I_{y2} \\ \vdots & \vdots \\ I_{xn} & I_{yn} \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} -I_{t1} \\ -I_{t2} \\ \vdots \\ -I_{tn} \end{bmatrix} \quad (5)$$

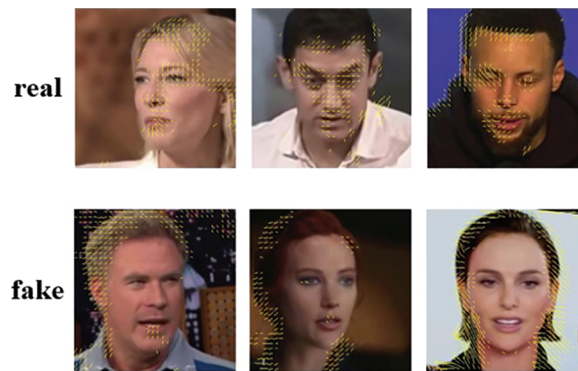
Eq. (5) can be abbreviated as  $A\vec{v} = -\vec{b}$ . The two unknowns can be solved for multiple equations by the least square method that multiplies the transpose of matrix  $A$  on both sides of Eq. (5), obtaining Eq. (6):

$$\vec{v} = (A^T A)^{-1} A^T (-\vec{b}) = \begin{bmatrix} \sum I_{xi}^2 & \sum I_{xi} I_{yi} \\ \sum I_{xi} I_{yi} & \sum I_{yi}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_{xi} I_{ti} \\ -\sum I_{yi} I_{ti} \end{bmatrix} \quad (6)$$

where  $i = 1, 2, \dots, m^2$ . According to Eq. (6), the product of the brightness difference of each pixel in the neighborhood of the  $x$ - and  $y$ -axes and that of the  $x$ -,  $y$ -, and  $t$ -axes is solved, and then the optical flow of the target pixel can be obtained by the matrix operation.

### 3.1.2 Differences in Optical Flow Features Between Real and Fake Faces

Deepfake videos have temporal features to be analyzed during the frame-by-frame playback. The differences between real and fake facial optical flow feature maps are shown in Fig. 3. The facial features of real videos will change to a large extent during the video playback and produce more optical flows in the center of the face. However, the organs of forged faces are stiff during the video playback, such that more optical flows will be generated at the interface between the face and the background. Therefore, the differences in the optical flow distribution between real and fake faces can be used as a basis for the classification.



**Figure 3:** Optical flow feature map difference between real and fake faces

### 3.1.3 Optical Flow Feature Extraction Strategies

To investigate the differences in optical flow image datasets formed by frame extraction strategies at different time intervals, this study uses the LK optical flow algorithm to extract optical flow

features of deepfake videos in three different approaches—every 2 frames, every 5 frames, and every 10 frames—at equal intervals. Fig. 4 displays the differences in the three strategies.



**Figure 4:** Differences in optical flow temporal features of different frame extraction strategies

Table 2 presents the differences in the characteristics and detection effects of different frame extraction strategies.

**Table 2:** Differences in characteristics and detection effects of different strategies

Strategy	Time interval	Characteristic	Detection effect
(1)	Two frames	The time interval between the adjacent frames is too short, and the range of the facial movement in the video is too small.	The extracted optical flows are insufficient, and only concentrated in the local area of the face, which cannot fully represent the temporal features of the face.
(2)	Five frames	The time interval between the adjacent frames is appropriate.	The extracted optical flows are smooth, independent, and focused in the center of the face, fully representing the variation on the temporal features of the faces.
(3)	Ten frames	The time interval between the adjacent frames is too long, and the range of the facial movement in the video is too large.	The extracted optical flows are overly concentrated at the junction of the person and the background. At the same time, the optical flows are too dense and intersect, which interferes with the classification.

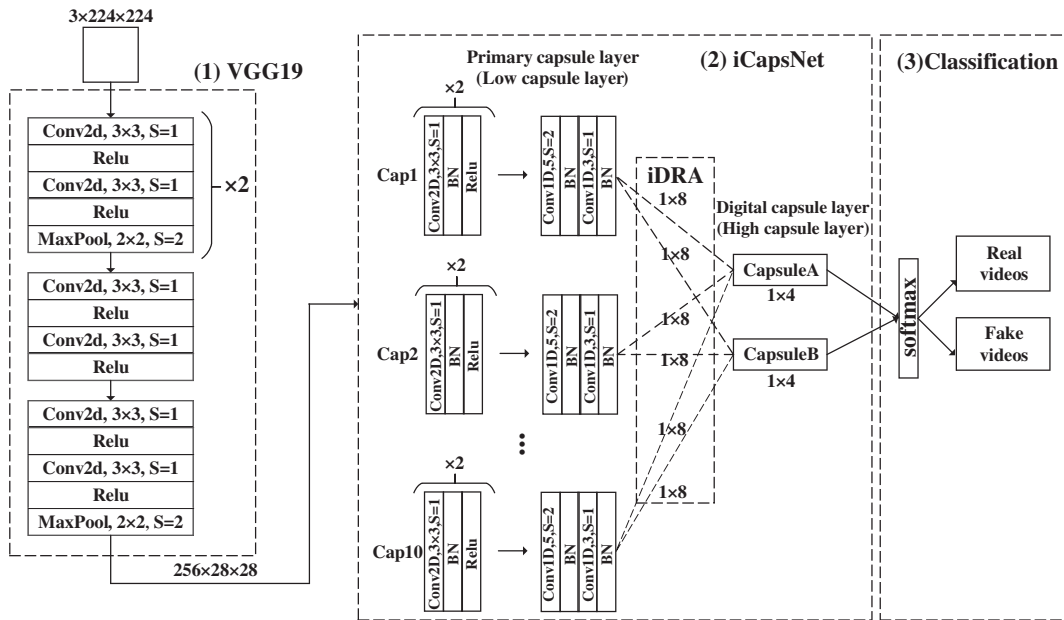
According to the explanation provided in Table 2, this study adopts Strategy (3) to obtain optical flow feature images and form a dataset of temporal features.

### 3.2 Capsule Network Training

The CNN extracts image features by stacking convolutional and pooling layers; however, neurons are represented in the form of the scalar quantity and lack the important direction attribute. Therefore, neurons cannot effectively represent the spatial position relationship of facial features. In addition to the differences in the spatial pixel features between the real and fake faces, the spatial position relationship of facial features must also be detected. Therefore, we employ the capsule network instead of the CNN to learn more spatial features of the faces.

#### 3.2.1 Detection Framework

Nguyen et al. [14] first applied the capsule network to the deepfake video detection. On the basis of the network structure proposed by Nyugen, we propose the iDRA by optimizing the dynamic routing algorithm, and the corresponding network is the iCapsNet. Fig. 5 presents the detection framework, which can be divided into three parts.



**Figure 5:** Detection framework

- (1) The VGG19 network helps in the preliminary extraction of facial image features. For the input facial image with a size of  $3 \times 224 \times 224$ , to reduce the number of subsequent parameters, the image features are first extracted via one part of the VGG19 network (composed of eight convolution layers, three maximum pooling layers, and eight ReLU functions). Then, a  $256 \times 28 \times 28$  feature map is output.
- (2) The iCapsNet method helps transform low-level specific features to high-level abstract features through the iDRA and iCapsNet. The feature map is input into the capsule network. First, ten primary capsules are used to further extract facial spatial features from different angles. Second, 2D and 1D convolutional layers are used to form vector neurons that represent facial image features from different directions. Third, the positional relationship between these neurons is explored. Then, the facial spatial features are transformed into a  $10 \times 8$ -dimensional



vector. Finally, the iDRA generates a  $2 \times 4$ -dimensional digital capsule with weight updates and the feature abstract representation.

- (3) The classification function helps classify true and false faces. The  $2 \times 4$ -dimensional vector is used as the preliminary basis for the classification. Then, the softmax function converts the value of each element of the feature vector into the interval (0, 1), and the sum of all elements is 1, as shown in Eq. (7).

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{c=1}^C e^{z_c}} \quad (7)$$

Facial images pass through these three parts in turn, and finally the framework distinguishes them.

### 3.2.2 Dynamic Routing Algorithm

As the core algorithm of the capsule network, the DRA [31] is an important method for a low-layer capsule to transmit information to a high-layer capsule. The algorithm essentially updates the weight between the low- and high-layer capsules through the dot product operation of the vector neurons and an increase in the number of iterations, such that the low-layer capsule is accurately clustered into the high-layer capsule, and the specific characteristics are transformed into abstract characteristics.

Table 3 shows the DRA, where  $\mathbf{u}_{ji}$  represents low-layer capsules,  $\mathbf{v}_j$  represents high-layer capsules,  $r$  represents the number of dynamic routing iterations,  $\mathbf{W}_{ij}$  represents a transformation matrix and updates parameters through model training,  $b_{ij}$  is the weight between the high- and low-layer capsules before using the softmax function,  $c_{ij}$  is the weight between the high- and low-layer capsules after using the softmax function. The squash function is the compression function.

$$\mathbf{u}_{ji} = \mathbf{W}_{ij} \mathbf{u}_i \quad (8)$$

$$\mathbf{v}_j = \frac{\|\mathbf{s}_j\|^2}{1 + \|\mathbf{s}_j\|^2} \frac{\mathbf{s}_j}{\|\mathbf{s}_j\|} \quad (9)$$

$$\mathbf{u}_{ji} \cdot \mathbf{v}_j = \|\mathbf{u}_{ji}\| \|\mathbf{v}_j\| \cos \theta \quad (10)$$

The procedure of the DRA is as follows. (1) As in Eq. (8), the dimension of the low-layer capsule is changed by multiplying  $\mathbf{u}_i$  with matrix  $\mathbf{W}_{ij}$  to obtain  $\mathbf{u}_{ji}$ , and the weight  $b_{ij}$  is initialized. (2) A softmax function is performed on  $b_{ij}$  and outputs the new weight  $c_{ij}$ , which is multiplied by the corresponding low-layer capsules and then summed as the first clustering result of the high-layer capsules. As shown in Eq. (9), the length of high-layer capsules is limited to a specific bounded interval. (3) A dot product operation is carried out between the first clustering result and low-layer capsules. The squash function is used to ensure that the direction of the high-layer capsules does not change, the length lies within the interval (0, 1), and the value of  $\mathbf{v}_j$  is updated according to the result of the dot product operation.

Through  $r$  iterations,  $b_{ij}$  becomes convergent and finally completes the update of the low-layer capsule to the high-layer capsule. Eq. (10) shows the dot product operation. If the length of the two vectors is longer, the angle between them will become smaller, and the result of the dot product operation will become larger.

**Table 3:** Dynamic routing algorithm (DRA)**Algorithm 1:** Dynamic routing algorithm**Input:**  $u_{j|i}, r, l$ **Output:**  $v_j$ 


---

```

1: for each capsule  $i$  in layer  $l$  and each capsule  $j$  in layer  $(l + 1)$ 
2:    $b_{ij} \leftarrow 0$ 
3: for  $r$  iterations
4:   for each capsule  $i$  in layer  $l$ 
5:      $c_i \leftarrow \text{softmax}(b_i)$ 
6:   for each capsule  $j$  in layer  $(l + 1)$ 
7:      $s_j \leftarrow \sum_i c_{ij} u_{j|i}$ 
8:      $v_j \leftarrow \text{squash}(s_j)$ 
9:   for each capsule  $i$  in layer  $l$  and each capsule  $j$  in layer  $(l + 1)$ 
10:     $b_{ij} \leftarrow b_{ij} + u_{j|i} \cdot v_j$ 
11: return  $v_j$ 

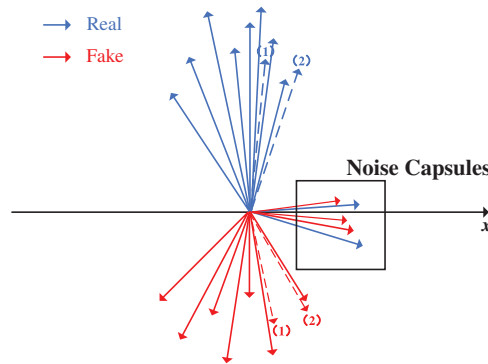
```

---

**3.2.3 Improved Dynamic Routing Algorithm**

The original DRA has two shortcomings when detecting a deepfake video.

First, when low-layer capsules are used to represent facial images, there are some similar feature vectors called noise capsules, as shown near the x-axis in Fig. 6. The dotted line at (1) in Fig. 6 represents the clustering center that must be obtained without the influence of noise capsules. However, low-layer capsules in the routing iteration process will be affected by noise capsules, because the low-layer capsules have been assigned the same weight as the noise capsules when being initialized, resulting in the deviation of the final clustering center, as shown by the dotted line at (2) in Fig. 6.

**Figure 6:** Influence of a noise capsule on clustering results

Second, on the basis of the dot product of the capsule and the initial clustering center, the original algorithm updates the weight of each low-layer capsule during dynamic routing iterations. For the noise capsules with a negative dot product of the initial clustering center, even though their weight ratio gradually decreases, they can continue to participate in the subsequent iterative process. Thus, the final clustering effect is not accurate, and the model is unable to effectively distinguish forged faces.

This study improves the original dynamic routing algorithm and proposes the iDRA to address the two shortcomings separately.

In view of the first shortcoming, the method of calculating the text similarity in the field of natural language processing is used to preliminarily estimate the similarity between low-layer capsules in the same layer.

Before the routing iteration, the cosine similarity between the low-layer capsules in the same layer is calculated and summed. Let  $\mathbf{u}_{j|i}$  be a low-layer capsule and  $\mathbf{u}_{j|k}$  be the other capsule of the same layer ( $k = 1, 2, \dots, 10$ ). Eq. (11) is used to calculate the cosine similarity between  $\mathbf{u}_{j|i}$  and the corresponding  $\mathbf{u}_{j|k}$ , the sum of which is assigned to  $F_i$ , as shown in Eq. (12). If the angle between the two vectors is less than  $90^\circ$ , their direction will be consistent, indicating their greater similarity. If the angle between the two vectors is larger than  $90^\circ$ , their direction will be opposite, indicating their lesser similarity. For noise capsules, the result of cosine similarity with most capsules is negative, and the value of  $F_i$  is small. Therefore, the noise capsules can be distinguished.

$$\cos(\mathbf{u}_{j|i}, \mathbf{u}_{j|k}) = \frac{\mathbf{u}_{j|i} \cdot \mathbf{u}_{j|k}}{\|\mathbf{u}_{j|i}\| \|\mathbf{u}_{j|k}\|} \quad (11)$$

$$F_i = \sum_{k=1}^n \cos(\mathbf{u}_{j|i}, \mathbf{u}_{j|k}) \quad (12)$$

After calculating the values of all low-layer capsules, the capsule  $\mathbf{u}_{j|i'}$  corresponding to the maximum  $F_i$  is selected as the benchmark capsule, as shown in Eq. (13).

$$\mathbf{u}_{j|i'} = \operatorname{argmax}(F_i) \quad (13)$$

To reduce the influence of noise capsules on the initialization center selection, a softmax function is performed on the cosine similarity values of the other capsules with this benchmark capsule, and the result is used as the initialization weight of each low-layer capsule. The softmax function is calculated as shown in Eq. (14), where  $C$  is the number of low-layer capsules in the same layer, and  $C = 10$ .

$$\operatorname{softmax}[\cos(\mathbf{u}_{j|i}, \mathbf{u}_{j|i'})] = \frac{e^{\cos(\mathbf{u}_{j|i}, \mathbf{u}_{j|i'})}}{\sum_{c=1}^C e^{\cos(\mathbf{u}_{j|c}, \mathbf{u}_{j|i'})}} \quad (14)$$

Eq. (15) represents the improved initial weight assignment method. Unlike the original DRA, Eq. (15) assigns different weights to different low-layer capsules, which helps improve the final clustering effect.

$$b_{ij} = \operatorname{softmax}[\cos(\mathbf{u}_{j|i}, \mathbf{u}_{j|i'})] \quad (15)$$

The different initial weights are multiplied by the corresponding low-layer capsules to obtain the initial clustering center of the high-level capsules  $\mathbf{s}_j$ , as shown in Eq. (16).

$$\mathbf{s}_j = \sum_i b_{ij} \mathbf{u}_{j|i} \quad (16)$$

In view of the second shortcoming, the weight of capsules with a positive dot product of the initial clustering center remains unchanged in the routing iteration process, while the weight of the noise capsules with a negative dot product is assigned to zero, as shown in Eq. (17).

$$\mathbf{u}_{j|i} \cdot \mathbf{v}_j = \begin{cases} \mathbf{u}_{j|i} \cdot \mathbf{v}_j, & \mathbf{u}_{j|i} \cdot \mathbf{v}_j > 0 \\ 0, & \mathbf{u}_{j|i} \cdot \mathbf{v}_j \leq 0 \end{cases} \quad (17)$$

Simultaneously, the sum of all dot product results in Eq. (17) is saved in the intermediate variable  $m_j$ . In fact,  $m_j$  only contains the dot product results with positive values. The weights between high- and

low-layer capsules are redistributed according to the ratio of the result of Eq. (17) to  $m_j$ , so as to reduce the effect of noise capsules. Thus, the final clustering result is more accurate, as shown in Eq. (18). Furthermore, in the subsequent routing iteration process, only the low-layer capsules consistent with the direction of the clustering center are considered, which can appropriately reduce the calculation amount.

$$b_{ij} = \frac{\mathbf{u}_{j|i} \cdot \mathbf{v}_j}{m_j} \quad (18)$$

Table 4 presents the improved dynamic routing algorithm.

**Table 4:** Improved dynamic routing algorithm (iDRA)

---

**Algorithm 2:** Improved dynamic routing algorithm

---

**Input:**  $\mathbf{u}_{j|i}, r, l$

**Output:**  $\mathbf{v}_j$

```

1: for each capsule  $i$  in layer  $l$  and each capsule  $j$  in layer  $(l + 1)$ 
2:    $F_i \leftarrow \sum_{k=1}^n \cos(\mathbf{u}_{j|i}, \mathbf{u}_{j|k})$ 
3:    $\mathbf{u}_{j|i'} \leftarrow \arg\max(F_i)$ 
4:    $b_{ij} \leftarrow \text{softmax}[\cos(\mathbf{u}_{j|i}, \mathbf{u}_{j|i'})]$ 
5: for  $r$  iterations
6:   for each capsule  $j$  in layer  $(l + 1)$ 
7:      $\mathbf{s}_j \leftarrow \sum_i b_{ij} \mathbf{u}_{j|i}$ 
8:      $\mathbf{v}_j \leftarrow \text{squash}(\mathbf{s}_j)$ 
9:   for each capsule  $i$  in layer  $l$  and each capsule  $j$  in layer  $(l + 1)$ 
10:     $m_j \leftarrow 0$ 
11:    if  $\mathbf{u}_{j|i} \cdot \mathbf{v}_j > 0$ 
12:       $\mathbf{u}_{j|i} \cdot \mathbf{v}_j \leftarrow \mathbf{u}_{j|i} \cdot \mathbf{v}_j$ 
13:    else  $\mathbf{u}_{j|i} \cdot \mathbf{v}_j \leftarrow 0$ 
14:     $m_j \leftarrow m_j + \mathbf{u}_{j|i} \cdot \mathbf{v}_j$ 
15:   for each capsule  $i$  in layer  $l$  and each capsule  $j$  in layer  $(l + 1)$ 
16:     $b_{ij} \leftarrow \frac{\mathbf{u}_{j|i} \cdot \mathbf{v}_j}{m_j}$ 
17: return  $\mathbf{v}_j$ 

```

---

### 3.3 Classification of Real and Fake Faces

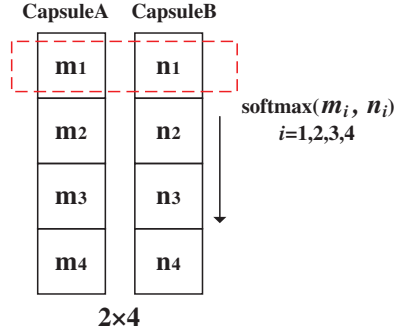
#### 3.3.1 Comprehensive Decision Function

The iDRA transforms the  $10 \times 8$ -dimensional low-layer capsule into a  $2 \times 4$ -dimensional high-layer capsule, which can be used to distinguish the feature vectors of real and fake faces. The spatial feature vectors and optical flow temporal feature vectors are assigned different weights to obtain the temporal-spatial feature decision function, as shown in Eq. (19).

$$ST_{final} = \lambda S_i + (1 - \lambda) T_i \quad (19)$$

### 3.3.2 Classification Basis

For the generated  $2 \times 4$ -dimensional capsules, each component value is positive or negative. The softmax function is used to reintegrate the corresponding component values of the high-layer capsules A and B, which are limited to the specific interval  $[0, 1]$ , as shown in Fig. 7.



**Figure 7:** Reintegration of capsule components based on function

Specifically, when the number of  $m_i$  components is two more than that of  $n_i$  components, it is determined to be a real face. Otherwise, it is determined to be a fake face, as shown in Eq. (20).

$$\text{result} = \begin{cases} \text{Real, } \text{count}(m_i > n_i) > 2 \\ \text{Fake, } \text{count}(m_i > n_i) \leq 2 \end{cases} \quad (20)$$

## 4 Experimental Results and Analysis

### 4.1 Datasets

We selected the Celeb-DF dataset [32] and the FaceForensics ++ dataset [33] for our experiments. The Celeb-DF dataset was jointly released by the University of Albany in New York and the University of the Chinese Academy of Sciences. It consists of 590 real videos and 5,639 fake videos of celebrities of different ages, genders, and regions on YouTube. The FaceForensics ++ dataset selected 1,000 public facial videos on YouTube to generate fake videos by FaceSwap, Deepfakes, Fac2Face, and NeuralTextures. We selected the experimental samples corresponding to FaceSwap and Deepfakes. Some samples of the datasets are shown in Fig. 8.



**Figure 8:** Samples of datasets

#### 4.2 Hyperparameter Settings

To reduce the overfitting phenomenon of the model, dropout was used to randomly discard some neurons. Other specific parameter settings are shown in Table 5.

**Table 5:** Hyperparameter settings

Parameter	Value
Epoch	20
Batch size	32
Number of primary capsules	10
Number of digital capsules	2
Number of dynamic routing iterations	3
Dropout rate	0.01
Learning rate	$5 \times 10^{-3}$

#### 4.3 Evaluation Index

The following evaluation indexes are selected to comprehensively evaluate the performance of the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (21)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (22)$$

$$\text{Recall} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (23)$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (24)$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (25)$$

$$L(\theta_0, \theta_1) = \text{loss}(x, y, \theta) = -\frac{1}{m} \sum_{i=1}^m y_i \log(q(x_i)) \quad (26)$$

The accuracy represents the probability that the number of correctly predicted samples accounts for the total sample size, as shown in Eq. (21). The precision represents the probability that a sample predicted to be a real face is actually a real face, as shown in Eq. (22). The recall rate is the true positive rate (TPR), which represents the probability that a sample that is actually a real face is predicted to be a real face, as shown in Eq. (23). The F1 value represents the comprehensive decision of the precision and the recall rate, as shown in Eq. (24). The false positive rate (FPR) represents the probability that a sample predicted to be a real face is actually a fake face, as shown in Eq. (25).

Furthermore, the cross-entropy loss function is used to evaluate the effectiveness of the model classification results, as shown in Eq. (26), where  $y_i$  is the actual category of the sample  $x_i$ ,  $q(x_i)$  is the



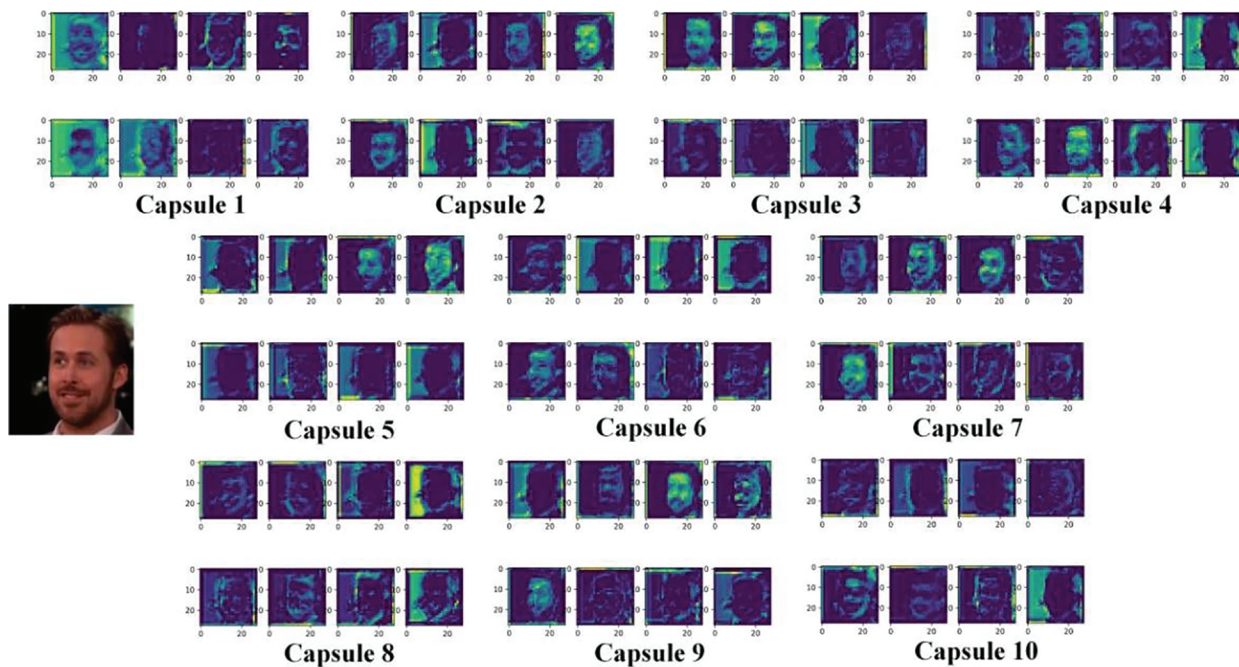
prediction probability of the sample  $x_i$ ,  $m$  is the number of samples, and  $\theta_0$  and  $\theta_1$  are the weights and bias values of the model, respectively.

#### 4.4 Analysis of Experimental Results

To verify the effectiveness of the method proposed in this study, eight comparative analysis experiments are set up in this section, seven of which are conducted on the Celeb-DF dataset with a higher forgery quality. At the end, we compared the detection accuracy of the proposed method on the Celeb-DF, FaceSwap, and Deepfakes datasets.

##### 4.4.1 Feature Map Extraction Experiment

To verify that different primary capsules are capable of extracting different facial features, the middle-layer feature maps of the ten primary capsules were extracted, as shown in Fig. 9.



**Figure 9:** Different facial feature maps extracted from different capsules

Fig. 9 shows that some capsules are able to focus on the core areas of the face, such as the eyes, nose, and mouth, while others focus on the marginal areas of the face, such as the forehead and the areas between the entire face and the background. Different capsules focus on different regions, such that different spatial features can be extracted. At the same time, each capsule can compensate for the key features not extracted by the other, ensuring that the spatial features extracted by the model are more comprehensive.

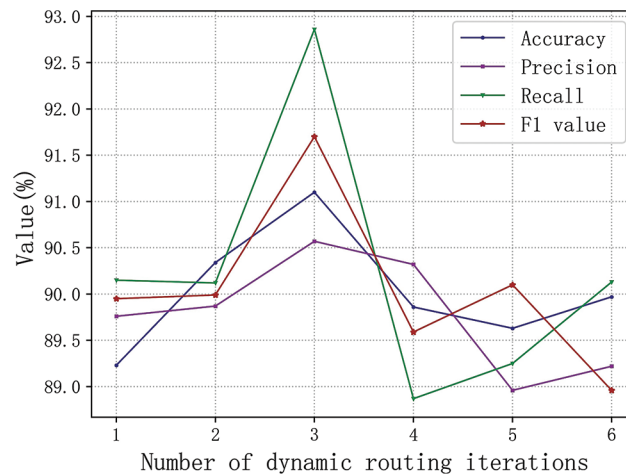
##### 4.4.2 The Comparative Experiment of Routing Iterations

This experiment is based on the original DRA without fusing optical flow features and is conducted by setting different numbers of routing iterations. The evaluation indexes include the

accuracy, precision, recall rate, and F1 value. The experimental results are displayed in Table 6 and Fig. 10.

**Table 6:** Influence of iteration number on classification effect

Number of iterations	Accuracy (%)	Precision (%)	Recall (%)	F1 value (%)	Cross-entropy loss
1	89.23	89.76	90.15	89.95	0.2187
2	90.34	89.87	90.12	89.99	0.2156
3	<b>91.10</b>	<b>90.57</b>	<b>92.86</b>	<b>91.70</b>	<b>0.2023</b>
4	89.86	90.32	88.87	89.59	0.2235
5	89.63	88.96	89.25	90.10	0.2213
6	89.97	89.22	90.13	88.96	0.2374



**Figure 10:** Classification effect trend with number of iterations

As shown in Fig. 10, the number of iterations of the DRA has an impact on the classification effects.

First, as the number of iterations increases, the accuracy and other indexes gradually increase, indicating that the capsule network gradually extracts more spatial features from the faces. This process can be approximated as a clustering process.

As the number of iterations increases, final detection results improve. When the number is set to three, all evaluation indexes reach the maximum value. As the number continues to increase, the classification accuracy begins to decrease, indicating that the update process of dynamic routing is not exactly equivalent to the clustering process. Therefore, in the subsequent experiments, the number of dynamic routing iterations  $r$  is set to three to obtain better classification results.

#### 4.4.3 Comparative Experiment of Frame Extraction Method

On the basis of the original DRA, this experiment sets the interval number of temporal frames to 2, 5, and 10 and the interval number of spatial frames to 5, 10, and 15 to verify the influence of

different frame extraction combinations on the detection accuracy and the F1 value. Table 7 displays the experimental results.

**Table 7:** Detection effects of different frame extraction methods

Method	Number of frame intervals for temporal features	Number of frame intervals for spatial features	Accuracy (%)	F1 value (%)
(1)	2	5	87.85	87.98
(2)	2	10	91.53	91.15
(3)	2	15	86.37	85.46
(4)	5	5	89.24	89.56
(5)	5	10	<b>92.76</b>	<b>91.87</b>
(6)	5	15	87.32	87.03
(7)	10	5	86.87	87.22
(8)	10	10	88.74	89.82
(9)	10	15	85.93	86.36

The comparison among experimental results of Methods (2), (5), and (8) shows that the different temporal feature extraction methods have different impacts on the detection accuracy when the frame intervals are maintained the same for spatial features. Method (5), using five-frame intervals, can best extract the smooth and independent optical flow features in the center of the faces. Compared with Method (2) using two-frame intervals, and Method (8) using ten-frame intervals, Method (5) has increasing accuracy and F1 value, which indicates that the number of frame intervals for temporal features cannot be too long or too short.

The comparison among the experimental results of Methods (4), (5), and (6) shows that Method (5) is capable of fully learning the spatial features of the faces when using five-frame intervals. As for Method (4), due to the small time span of adjacent frames, the variation range of facial actions and expressions is not large, resulting in the spatial feature redundancy and overfitting of the model. Method (6) cannot obtain sufficient spatial feature samples due to the large time span of adjacent frames, both of which result in decreased detection effect.

Moreover, for Method (5), although the detection effect is the best, video frames must be extracted twice at different intervals, which consumes excessive time in the data processing. Therefore, in practical application, if numerous faked videos must be detected within a short time, we can choose Method (4) to process the video data; if it is necessary to achieve higher detection accuracy, we choose Method (5).

#### 4.4.4 Comparative Experiment of Temporal–Spatial Feature Fusion

In this experiment, spatial and temporal features are learned separately by the original DRA. The classification effects of different fusion strategies are compared by changing the value of  $\lambda$ . The experimental results are shown in Table 8, where  $\lambda$  is the weight of spatial features and  $1 - \lambda$  is the weight of temporal features.

**Table 8:** Classification effects of different fusion strategies

$\lambda$	$1 - \lambda$	Accuracy (%)	F1 value (%)	Cross-entropy loss
0	1	87.95	87.71	0.2467
0.1	0.9	88.23	88.56	0.2253
0.2	0.8	88.67	87.82	0.2135
0.3	0.7	89.51	88.41	0.2096
0.4	0.6	90.12	90.34	0.2052
0.5	0.5	90.87	89.93	0.2078
0.6	0.4	91.25	91.46	0.1987
0.7	0.3	92.22	<b>91.73</b>	0.1931
0.8	0.2	<b>92.54</b>	91.53	<b>0.1902</b>
0.9	0.1	91.83	91.37	0.1956
1	0	91.15	91.70	0.2023

The following conclusions can be drawn from Table 8.

- (1) The classification accuracy reaches the highest when  $\lambda$  is 0.8,  $1 - \lambda$  is 0.2, and the F1 value is also close to its maximum.
- (2) When the  $\lambda$  value is 1, the classification effect is better than that when the  $\lambda$  value is 0. This indicates that the classification effect when using spatial features alone is better than that when using temporal features alone.
- (3) When the value of  $1 - \lambda$  is within the range of [0.1, 0.4], the classification accuracy and cross-entropy loss are better than those when the value of  $1 - \lambda$  is 0. This indicates that the classification effect is improved after the optical flow temporal features are fused, demonstrating the effectiveness of the temporal-spatial feature fusion method.
- (4) As the value of  $\lambda$  gradually increases from 0, and the value of  $1 - \lambda$  gradually decreases, the model classification effect improves, indicating that spatial features play a dominant role in the model classification. Therefore, the use of spatial features or optical flow temporal features alone for the classification has its own limitations, whereas a proper fusion of the two features can achieve the best detection effect. In the subsequent experiments, the value of  $\lambda$  is set to 0.8 to obtain better results.

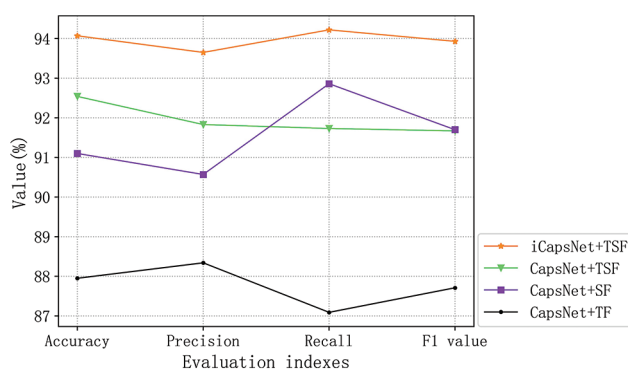
#### 4.4.5 Comparative Experiment of Each Improvement Strategy in Terms of Performance Gain

In this experiment, XceptionNet [34] is used as the benchmark model. The temporal features, spatial features, temporal-spatial features, and iDRA are gradually superimposed on CapsNet. The evaluation indexes include accuracy, precision, recall, F1 value, and cross-entropy loss. Table 9 and Fig. 11 present the results.

Both XceptionNet and CapsNet are used to learn spatial features. Because of the use of vector neurons and the DRA, CapsNet can give full consideration to the size and direction difference of spatial features of the face and achieve better detection results than XceptionNet.

**Table 9:** Classification performance gain generated by each improvement strategy

Detection method	Accuracy (%)	Precision (%)	Recall (%)	F1 value (%)	Cross-entropy loss
XceptionNet + SF	84.24	84.52	83.87	84.19	0.2735
CapsNet + TF	87.95	88.34	87.09	87.71	0.2378
CapsNet + SF	91.15	90.57	92.86	91.70	0.2023
CapsNet + TSF	92.54	91.83	91.69	91.73	0.1902
CapsNet + iDRA + TSF (iCapsNet + TSF)	<b>94.07</b>	<b>93.65</b>	<b>94.22</b>	<b>93.93</b>	<b>0.1769</b>

**Figure 11:** Classification effect changes with superposition of each improvement strategy

On the basis of CapsNet, temporal, spatial, and temporal-spatial features are superimposed one by one on CapsNet + TF, CapsNet + SF, and CapsNet + TSF. The comparison shows that the classification accuracy can be improved by 3.2% when using spatial features alone compared with using temporal features alone. When using the temporal-spatial features, though the recall rate declines, the classification accuracy is 1.6% higher than that using only spatial features. Besides, the F1 value and cross-entropy loss are optimized.

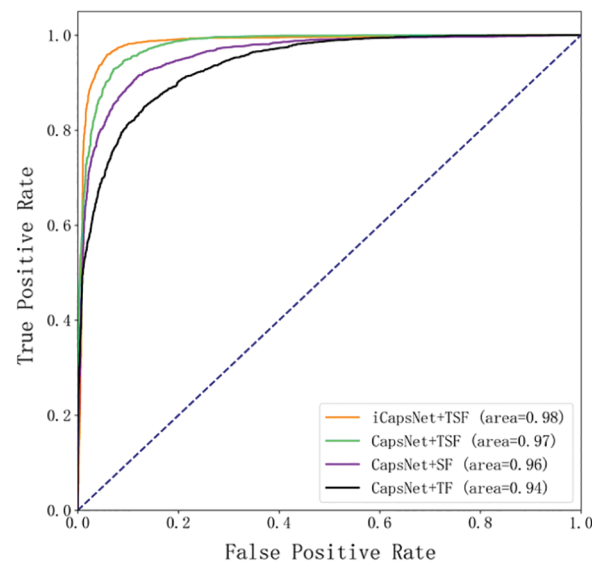
The comparison of CapsNet + TSF and iCapsNet + TSF shows that each evaluation index is improved owing to the iDRA of iCapsNet. The iDRA can enhance the final clustering effect by assigning different initial weights to each low-layer capsule and reducing the interference of noise capsules during the iterative process, such that the high-level capsule more accurately represents the differences in the real and fake facial images by direction and size.

Method (5) indicates that the use of both temporal-spatial features and the iDRA can lead to the best classification of deepfake videos, verifying the effectiveness of the proposed method.

#### 4.4.6 Receiver Operating Characteristic Curve and Confusion Matrix

The receiver operating characteristic (ROC) curve is used to further evaluate the effectiveness of the proposed method. By setting different thresholds, different TPR and FPR values are obtained. The ROC curve is drawn with FPR as the abscissa and TPR as the ordinate. Smoother ROC curves indicate lesser overfitting. The area under curve (AUC) indicates the area under the ROC curve, and its value range is [0.5, 1]. Higher AUC values indicate a better detection effect.

Each improvement strategy is gradually superimposed to draw the ROC curve, as shown in Fig. 12. With the gradual superposition of each improvement strategy, the ROC curve is gradually smoothed, and AUC is gradually increasing. Because iCapsNet employs the iDRA, the overfitting phenomenon during the model training is less than that of CapsNet. Furthermore, the detection effect when using spatial and temporal features simultaneously is better than that when using each alone, which verifies the effectiveness of the iCapsNet-TSF method proposed in this study.



**Figure 12:** ROC curve

Meanwhile, the confusion matrix of the iCapsNet-TSF detection method is drawn, as shown in Fig. 13. The brighter color represents larger sample numbers. Of the 5,900 samples to be detected, more than 5,500 samples are detected correctly. The number of samples where real faces are predicted as real faces—i.e., the true positive (TP) value—is 2,850. The number of samples where fake faces are predicted as fake faces—i.e., the true negative (TN) value—is 2,735. The number of samples where real faces are predicted as fake faces—i.e., the false positive (FP) value—is 123. The number of samples where fake faces are predicted as real faces—i.e., the false negative (FN) value—is 196. The TP and TN values are high, and FP and FN values are low, which once more proves the effectiveness of the proposed method.

#### 4.4.7 Comparative Experiment with Other Algorithms

This experiment selects other mainstream deepfake video detection methods for a comparison, including Durall R, MesoNet, the convolutional recurrent neural network (CRNN), and XceptionNet. Evaluation indexes include accuracy, F1 value, time consumed in one round of the model training and the number of parameters of the model. Table 10 presents the experimental results.

The comparison of Durall R and iCapsNet-TSF shows that the iCapsNet-TSF method can realize the training of massive datasets by the deep learning model. Hence, its detection effect is better than that of the machine learning algorithm. However, the number of parameters that the Durall R method uses is small, which saves more time.



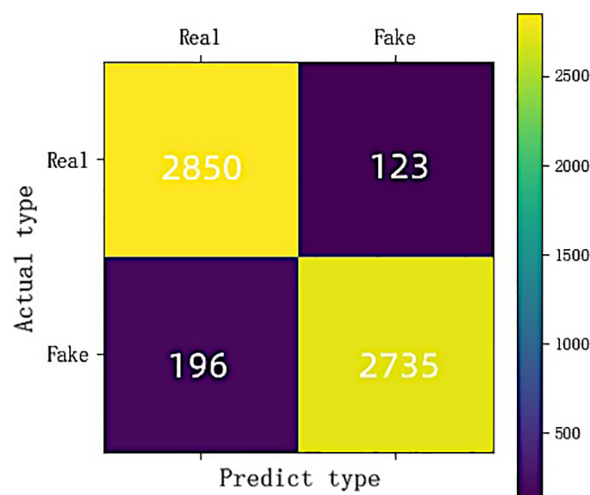


Figure 13: Confusion matrix

Table 10: Comparison with other algorithms

Detection method	Accuracy (%)	F1 value (%)	Time consumed (min)	Number of parameters
Durall R [10]	72.56	73.32	<b>5.46</b>	<b>233,540</b>
MesoNet [12]	79.13	78.67	9.12	9,323,430
CRNN [24]	77.45	77.82	7.43	5,567,250
XceptionNet [34]	84.24	84.19	15.37	47,835,330
iCapsNet-TSF	<b>94.07</b>	<b>93.93</b>	12.51	38,959,980

By comparing MesoNet, XceptionNet, and iCapsNet-TSF, MesoNet and XceptionNet are both CNN detection methods, and scalar neurons are used to represent facial features. In the iCapsNet-TSF method, the capsule network uses vector neurons to extract facial features to achieve better detection results. Furthermore, it consumes less time and has fewer parameters than XceptionNet.

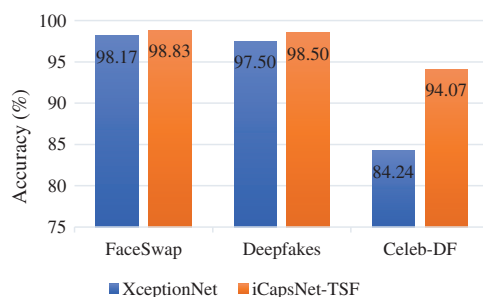
The comparison between CRNN and iCapsNet-TSF shows that the CRNN method focuses on the temporal variation in the spatial features of facial videos, whereas the iCapsNet-TSF method focuses on spatial features and the spatial variation in temporal features. The iCapsNet-TSF method is more effective, indicating that spatial features as the classification basis are more important than temporal features in the deepfake video detection.

#### 4.4.8 Comparative Experiment of Detection Effects on Different Datasets

Using XceptionNet as the benchmark model, we further compare the detection accuracy of the proposed method, iCapsNet-TSF, on the FaceSwap, Deepfakes, and Celeb-DF datasets.

As shown in Fig. 14, the detection accuracy of iCapsNet-TSF is slightly higher than that of XceptionNet for the FaceSwap and Deepfakes datasets, due to the low video quality in the FaceSwap and Deepfakes datasets and the slight overfitting phenomenon in the model learning process. However,

on the Celeb-DF dataset, XceptionNet is not as effective as the iCapsNet-TSF method, because it fails to fully extract the spatial and temporal features of the face.



**Figure 14:** Detection accuracy of proposed method on different datasets

iCapsNet uses vector neurons, which effectively retain detailed information on the position and posture of the detected object through the most important direction attributes. Therein, the improved DRA is used to fully transfer the information of vector neurons at different levels, and it hence performs better than XceptionNet. The advantages of the capsule network and temporal-spatial feature fusion strategy over traditional CNNs in the deepfake detection are demonstrated.

## 5 Conclusions

We propose a deepfake video detection method based on the fusion of improved CapsNet and temporal-spatial features. The iCapsNet-TSF method combines the capsule network and the optical flow algorithm to complete the deepfake detection for the first time. The optical flow algorithm fully extracts the temporal features of forged videos. iCapsNet makes a comprehensive decision based on temporal-spatial features by weight initialization and updating on a DRA, providing a novel strategy for the deepfake detection. The resulting detection accuracy is significantly improved. A deep learning model can be trained to obtain accurate judgments for a given deepfake video dataset; however, the accuracy often decreases when faced with a cross-dataset detection task. To address this problem and enhance the generalization ability and interpretability of the model, meta-learning, small-sample learning, and image segmentation theory can be introduced. Future studies must focus on improving the generalization of the capsule network. Meanwhile, the capsule network can be used in the fields of the target detection and the image segmentation, and the optical flow algorithm can be used to extract the features of other time-series data.

**Funding Statement:** This study is supported by the Fundamental Research Funds for the Central Universities under Grant 2020JKF101 and the Research Funds of Sugon under Grant 2022KY001.

**Conflicts of Interest:** The authors declare that they have no conflict of interest to report regarding the present study.

## References

- [1] L. Verdoliva, "Media forensics and deepFakes: An overview," *IEEE Journal on Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 910–932, 2020.
- [2] C. Cross, "Using artificial intelligence (AI) and deepfakes to deceive victims: The need to rethink current romance fraud prevention messaging," *Crime Prevention and Community Safety*, vol. 24, pp. 30–41, 2022.

- [3] M. A. Siddiqi, W. Pak and M. A. Siddiqi, "A study on the psychology of social engineering-based cyberattacks and existing countermeasures," *Applied Sciences*, vol. 12, pp. 1–19, 2022.
- [4] Y. Taher, A. Moussaoui and F. Moussaoui, "Automatic fake news detection based on deep learning, FastText and news title," *International Journal of Advanced Computer Science and Applications*, vol. 13, pp. 146–158, 2022.
- [5] F. Juefei-Xu, R. Wang, Y. Huang, L. Ma and Y. Liu, "Countering malicious deepfakes: Survey, battleground, and horizon," *International Journal of Computer Vision*, vol. 130, pp. 1678–1734, 2022.
- [6] M. Koopman, A. M. Rodriguez and Z. Geradts, "Detection of deepfake video manipulation," in *Proc. of the 20th Irish Machine Vision and Image Processing Conf. (IMVIP)*, Belfast, Northern Ireland, pp. 133–136, 2018.
- [7] Y. Luo, Y. Zhang, J. Yan and W. Liu, "Generalizing face forgery detection with high-frequency features," in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 16312–16321, 2021.
- [8] F. Matern, C. Riess and M. Stamminger, "Exploiting visual artifacts to expose deepfakes and face manipulations," in *Proc. of the 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA, pp. 83–92, 2019.
- [9] X. Yang, Y. Li and S. Lyu, "Exposing deepfakes using inconsistent head poses," in *Proc. of the 2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, pp. 8261–8265, 2019.
- [10] R. Durall, M. Keuper, F. J. Pfrendt and J. Keuper, "Unmasking deepfakes with simple features," arXiv:1911.00686v3, pp. 1–8, 2020.
- [11] D. Cozzolino, A. Rössler, J. Thies, M. Nießner and L. Verdoliva, "ID-Reveal: Identity-aware deepfake video detection," in *Proc. of the 2021 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 15108–15117, 2021.
- [12] D. Afchar, V. Nozick, J. Yamagishi and I. Echizen, "MesoNet: A compact facial video forgery detection network," in *Proc. of the 2018 IEEE Int. Workshop on Information Forensics and Security (WIFS)*, Hong Kong, China, pp. 1–7, 2018.
- [13] P. Zhou, X. Han, V. I. Morariu and L. S. Davis, "Learning rich features for image manipulation detection," in *Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, pp. 1053–1061, 2018.
- [14] H. H. Nguyen, J. Yamagishi and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos," in *Proc. of the 2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, pp. 2307–2311, 2019.
- [15] X. Zhu, H. Wang, H. Fei, Z. Lei and S. Z. Li, "Face forgery detection by 3D decomposition," in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 2928–2938, 2021.
- [16] C. Wang and W. Deng, "Representative forgery mining for fake face detection," in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 14918–14927, 2021.
- [17] J. Li, H. Xie, J. Li, Z. Wang and Y. Zhang, "Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection," in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 6458–6467, 2021.
- [18] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong et al., "Learning self-consistency for deepfake detection," in *Proc. of the 2021 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Montreal, QC, Canada, pp. 15023–15033, 2021.
- [19] X. Dong, J. Bao, D. Chen, T. Zhang, W. Zhang et al., "Protecting celebrities from deepfake with identity consistency transformer," in *Proc. of the 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, Louisiana, USA, pp. 9468–9478, 2022.
- [20] R. Shao, T. Wu and Z. Liu, "Detecting and recovering sequential deepfake manipulation," in *Proc. of the 2022 European Conf. on Computer Vision (ECCV)*, Tel-Aviv, Israel, pp. 1–22, 2022.

- [21] Z. Wang, Z. Wang, Z. Yu, W. Deng, J. Li *et al.*, “Domain generalization via shuffled style assembly for face anti-spoofing,” in *Proc. of the 2022 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, Louisiana, USA, pp. 4123–4133, 2022.
- [22] Q. Gu, S. Chen, T. Yao, Y. Chen, S. Ding *et al.*, “Exploiting fine-grained face forgery clues via progressive enhancement learning,” in *Proc. of the 2022 AAAI Conf. on Artificial Intelligence*, Vancouver, British Columbia, Canada, pp. 735–743, 2022.
- [23] K. Sun, T. Yao, S. Chen, S. Ding, J. Li *et al.*, “Dual contrastive learning for general face forgery detection,” in *Proc. of the 2022 AAAI Conf. on Artificial Intelligence*, Vancouver, British Columbia, Canada, pp. 2316–2324, 2022.
- [24] E. Sabir, J. Cheng, A. Jaiswal, W. Abdalmageed, I. Masi *et al.*, “Recurrent convolutional strategies for face manipulation detection in videos,” *Interfaces*, vol. 3, no. 1, pp. 80–87, 2019.
- [25] Y. Li, M. C. Chang and S. Lyu, “In ictu oculi: Exposing AI created fake videos by detecting eye blinking,” in *Proc. of the 2018 IEEE Int. Workshop on Information Forensics and Security (WIFS)*, Hong Kong, China, pp. 1–7, 2018.
- [26] I. Amerini, L. Galteri, R. Caldelli and A. D. Bimbo, “Deepfake video detection through optical flow based CNN,” in *Proc. of the 2019 IEEE/CVF Int. Conf. on Computer Vision Workshop (ICCVW)*, Seoul, Korea (South), pp. 1205–1207, 2019.
- [27] Z. Sun, Y. Han, Z. Hua, N. Ruan and W. Jia, “Improving the efficiency and robustness of deepfakes detection through precise geometric features,” in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 3608–3617, 2021.
- [28] A. Haliassos, K. Vougioukas, S. Petridis and M. Pantic, “Lips don’t lie: A generalisable and robust approach to face forgery detection,” in *Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA, pp. 5039–5049, 2021.
- [29] Z. Gu, Y. Chen, T. Yao, S. Ding, J. Li *et al.*, “Delving into the local: Dynamic inconsistency learning for deepfake video detection,” in *Proc. of the 2022 AAAI Conf. on Artificial Intelligence*, Vancouver, British Columbia, Canada, pp. 744–752, 2022.
- [30] J. L. Barron, D. J. Fleet and S. S. Beauchemin, “Performance of optical flow techniques,” *International Journal of Computer Vision*, vol. 12, pp. 43–77, 1994.
- [31] S. Sabour, N. Frosst and G. E. Hinton, “Dynamic routing between capsules,” in *Proc. of the 31st Neural Information Processing Systems (NIPS)*, Long Beach, CA, USA, pp. 3856–3866, 2017.
- [32] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, “Celeb-DF: A large-scale challenging dataset for deepfake forensics,” in *Proc. of the 2020 IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 3204–3213, 2020.
- [33] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies *et al.*, “FaceForensics ++: Learning to detect manipulated facial images,” in *Proc. of the 2019 IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 1–11, 2019.
- [34] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. of the 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 1251–1258, 2017.