Spatiotemporal Trident Networks: Detection and Localization of Object Removal Tampering in Video Passive Forensics

Quanxin Yang, *Student Member, IEEE,* Dongjin Yu, *Senior Member, IEEE,* Zhuxi Zhang, Ye Yao*, and Linqiang Chen

***Abstract*—With the development of video and image processing technology, the field of video tampering forensics is facing enormous challenges. Specifically, as the fundamental basis of judicial forensics, passive forensics for object removal video forgery is particularly essential. To extract tampering traces in video more sufficiently, the author proposed a spatiotemporal trident network based on the spatial rich model (SRM) and 3D convolution (C3D), which provides three branches and can theoretically improve the detection and localization accuracy of tampered regions. Based on the spatiotemporal trident network, a temporal detector and a spatial locator were designed to detect and locate the tampered regions in the temporal and spatial domains of videos. For the temporal detector, 3D CNNs were employed in three branches as the encoders and a bidirectional long short-term memory (BiLSTM) as the decoder. For the spatial locator, a backbone network named C3D-ResNet12 was designed as the encoder of the three branches, and the region proposal networks (RPNs) were employed as the decoders in three branches. In addition, we optimized the loss functions of the above two algorithms based on focal loss and GIoU loss. The experimental results revealed the effectiveness of spatiotemporal detection and localization algorithms: for temporal forgery detection, the accuracy of the frame classification increased to 99+%; for spatial forgery localization, the successful localization rate of the tampered regions in forged frames reached 96+%, and the mean intersection over union of the located tampered regions and the real tampered regions reached 62+%.**

***Index Terms*—Video passive forensics, object removal tampering, spatiotemporal localization, target detection based on 3D CNN, trident network.**

I. Introduction

With the updating of image and video editing software algorithms, it has become easier for people to maliciously forge a video that is difficult to distinguish from authentic video [1]. Once these videos are uploaded to a public network, they have a great impact on society [2]. Therefore, it is necessary to find fast and effective video authenticity identification methods.

Manuscript received August 15, 2020; revised November 14, 2020; accepted December 13, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 62071267 and in part by the Key Research and Development Project of Zhejiang Province China under Grant 2020C01165. *(Corresponding author: Ye Yao)*.

Q. Yang and D. Yu are with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China (yqx2018@hdu.edu.cn).

Z. Zhang, Y. Yao and L. Chen are with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China (yaoye@hdu.edu.cn).

Digital video is composed of visual objects with spatial structure and semantic information. Adding, deleting or modifying video objects often directly affects people's understanding of video content. Compared with forensic research on video forgery based on double compression, frame insertion, frame deletion and frame duplication, forensic research on video object forgery is more complex and meaningful [3]. However, adding or modifying moving objects in a video without leaving visible traces usually requires very professional and delicate operations. In contrast, as shown in Fig. 1, video object removal tampering is generally easier to operate with forgery traces invisible to human eyes. Therefore, we focus on discussing the passive video forensics of object removal forgery.

Digital video forensics technology includes two categories: active forensics and passive forensics. Active forensic technology usually needs to embed a priori information such as steganography [4] during video recording and then proofread the a priori information to ensure the authenticity of the video. Unfortunately, this technology remains a challenging task in many practical applications. In contrast, passive forensics do not rely on a priori information but only use a given digital video to determine whether it has been tampered with. However, research on passive video forensics is still in its infancy [1], and there is much room for exploration and improvement in forensic algorithms.

This paper proposed a spatiotemporal trident network based on SRM [5] and C3D [6], [7]. Moreover, based on the spatiotemporal trident network, two algorithms were designed to detect and locate the tampered regions of object removal video forgery in the temporal and spatial domains, respectively. First, we trained the tampered regions' temporal detector and spatial locator. The temporal detector was then used to detect the sequence of forged frames within the video. Finally, the spatial locator was employed to locate the tampered region within each forged frame. The main contributions of this paper are as follows:

1) To improve the accuracy of detection and localization of the tampered regions when inputting five consecutive frames, we proposed a spatiotemporal trident network and analyzed the theoretical basis of the accurate classification and localization of the middle frame.
2) According to the theoretical basis of the analysis, we designed a frame classification algorithm based on the spatiotemporal trident network. BiLSTM [8], [9]

Fig. 1: An example of video object removal tampering. The upper row shows the frames of an original video, while the lower row shows the frames of a forged video. The persons in the red circle are removed in the forged video frames.

was employed to learn the middle frame classification mechanisms. To further improve the classification accuracy of hybrid-frame input cases, we introduced the hybrid-frame weight for those cases as an extra penalty term in the loss function.

3) Based on the spatiotemporal trident network and combined with the RPN [10], [11], a spatial localization algorithm was designed to locate the tampered region within each forged frame. Furthermore, we designed a backbone network named C3D-ResNet12 [12] and a loss function for the spatial localization.

The rest of the paper is organized as follows. After discussing the related work in Sec. II, we introduce our spatiotemporal trident network in Sec. III and analyze its role in the temporal detection and spatial localization of video tampered regions. According to the theoretical basis obtained in Sec. III, the specific implementation of the temporal forgery detection algorithm is explained in Sec. IV, and the spatial forgery localization algorithm is described in Sec. V. After that, we give the experimental analysis of the two algorithms in Sec. VI. Finally, this paper is summarized in Sec. VII.

II. Related works

The detection and localization of object removal tampering in digital videos is a new research direction in digital video passive forensics. We summarize the existing video forensics algorithms in the following four categories.

1) Algorithms based on abstract statistical features: Saddique et al. [13] used descriptors to encode the different characteristics of consecutive frames; then, they used a support vector machine (SVM) to determine whether the consecutive frames have been tampered with. Chen et al. [3] collected some statistical features and input them into an SVM as feature vectors to classify natural objects and fake objects. These types of artificial acquisition and coding features are often relatively single, and valuable features may be difficult to extract sufficiently.

2) Algorithms based on noise patterns: Chen et al. [1] used a feature extractor for static image steganalysis to extract tampering features from motion residuals and proposed an object-based video forgery recognition algorithm based on a frame operation detector, and they provided a two-stage automatic algorithm to locate fake video clips in suspicious videos. Tan et al. [14] used a feature extractor built for image steganalysis in the frequency domain to extract tampering features from motion residuals and developed a method for automatic recognition of object-based forgery video using advanced frame coding based on the GOP structure. These noise features are generally visible to the naked eye, but these artificial algorithms still need to make constant attempts to quantify the feature and make a valid judgment.

3) Algorithms based on pixel correlation: Sitara and Mehtre [15] proposed a method for distinguishing optical camera zoom and synthetic zoom for video tampering detection based on pixel difference correlation and sensor pattern noise. Liu et al. [16] used brightness and contrast as local features to measure the similarity between foreground and background and then detected forgery by identifying these features' inconsistencies between foreground and background. This kind of algorithm is generally not suitable for video object removal tampering because the tampered patch has a great pixel correlation with the background frame.

4) Algorithms based on video content characteristics: Damiano et al. [17] proposed an algorithm for reliable detection and localization of video copy-move forgery based on image block matching. Aloraini et al. used spatial decomposition, temporal filtering, and sequence analysis in [18] to detect object-based video forgery and estimate the motion of the removed object; in [2], a method based on sequence analysis and patch analysis was proposed, which was used to detect object removal tampering and locate the tampered regions in the video. Zhong et al. [19] extracted multidimensional dense moment features from the video, then used the best interframe matching algorithm to identify the interframe forged video, and finally indicated the corresponding interframe tampered regions. The accuracy of this kind of artificial algorithm depends on the thresholds. Thus, improper selection of thresholds may easily result in incorrect matching or analysis.

Video passive forensics algorithms based on deep learning should generally be classified as noise pattern algorithms because almost all of them use video frame high-frequency information. Yao et al. [20] used the frame differential method to construct the frame difference sequence, extracted the high-

frequency signals of the frame difference sequence through a high-pass filter, and finally, used CNN to learn to distinguish the forged frames. Kohli et al. [21] proposed a spatiotemporal detection method based on CNN to detect and locate the tampered regions in the forged frames, and the motion residual was employed in the network. Adobe's dual-stream [5] Faster R-CNN was designed to detect the splicing, copying, and removing tampering regions in images. However, deep learning algorithms for video passive forensics have not yet sufficiently utilized the correlation and continuity features between consecutive frames.

In conclusion, to sufficiently extract the continuous and relevant features of tampering traces in the video temporal and spatial domains, we employ SRM to extract the video noise stream signals and employ 3D CNN to extract the depth features of tampering traces in the video noise stream. To improve the accuracy of 3D CNN in the input cases of hybrid frames, we propose a spatiotemporal trident network and design a temporal detection algorithm and a spatial localization algorithm for the tamped regions. In recent years, several papers research on video forensics have been published on IEEE TCSVT [1], [2], [17], [22]. The papers that are the most relevant to our paper are [1] and [2], which used the same dataset as our paper. Chen et al. [1] created the SYSU-OBJFORG dataset, introduced the issue of video object tampering detection, and proposed a temporal forgery detection algorithm based on motion residuals. Aloraini et al. [2] proposed an algorithm that can visualize the tampered area based on sequence analysis and patch analysis. While [17] and [22] focused on detecting video copy-move and video frame deletion, respectively. Our algorithms based on the proposed spatiotemporal trident network can automatically and sufficiently extract spatiotemporal features from videos. Therefore, our algorithms rarely need to define what features to extract and set many thresholds manually. Our algorithms can also detect and locate tampered regions in video frames more accurately in both temporal and spatial domains.

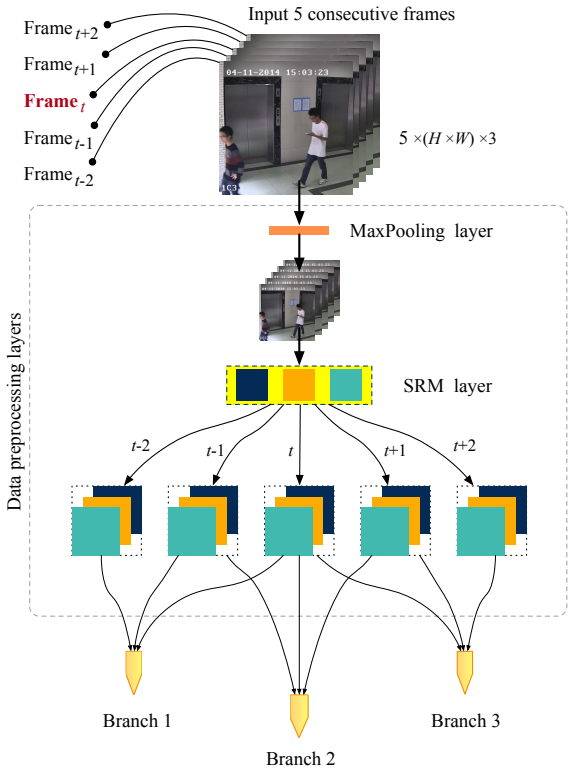


Fig. 2: Model structure of our spatiotemporal trident network. We use a slicing operation to divide the video noise stream into three branches, and each branch contains three consecutive frames' noise data.

$$1/4\times\begin{bmatrix}0&0&0&0&0\\0&-1&2&-1&0\\0&2&-4&2&0\\0&-1&2&-1&0\\0&0&0&0&0\end{bmatrix}\quad 1/12\times\begin{bmatrix}-1&2&-2&2&-1\\2&-6&8&-6&2\\-2&8&-12&8&-2\\2&-6&8&-6&2\\-1&2&-2&2&-1\end{bmatrix}\quad 1/4\times\begin{bmatrix}0&0&0&0&0\\0&0&-1&0&0\\0&-1&4&-1&0\\0&0&-1&0&0\\0&0&0&0&0\end{bmatrix}$$

Fig. 3: Three convolution kernels with fixed parameters.

III. Proposed Method

A. Spatiotemporal trident network

We proposed a spatiotemporal trident network shown in Fig. 2, which was designed as the preprocessing layers in spatiotemporal localization algorithms of the removed regions in video frames. Our method is suitable for videos captured by static cameras or videos with slow lens movement because suitable patches of tampered regions may be found in adjacent frames of such videos. The operation process of video object removal tampering generally includes four steps: 1) Decompress the original video into a sequence of frames; 2) Delete the target object and its surrounding pixels frame by frame; 3) Select the background patch from the adjacent frames to fill the deleted pixels; 4) Recompress the manipulated frame sequence into a video. Our goal is to detect the sequence of forged frames in the forged video and locate the patched region within each forged frame. 3D convolution is mostly used in the field of video content detection [23]–[25], which can extract the continuity features in adjacent frames to identify what type of event is occurring in the video. Generally, the input data unit for 3D CNN is the same category of clips in 3D space. However, in the forged video, there may be only several clips of consecutive frames that are forged. The input data unit may contain both pristine frames and forged frames at the junction of pristine frames and forged frames, while 3D CNN will lose the recognition ability of this input case. To detect every single frame and locate the tampered region for each forged frame based on 3D CNN, we propose a spatiotemporal trident network shown in Fig. 2. The idea of our algorithms is to obtain the prediction result of a single frame through consecutive multiple frames of information. Corresponding to the multiframe input to the middle frame is the most reasonable choice; thus, an odd number of frames should be selected. Compared with three frames and seven frames, five frames contain the most appropriate temporal domain information, and the network design is also very convenient. Specifically, in this paper, we set each of five consecutive frames as an input unit, where every three consecutive frames

are one branch to obtain three branches. The three branches' outputs are used as the input data streams for the temporal and spatial localization algorithms of the middle frame in five consecutive frames.

Both the temporal and spatial localization algorithms employ the spatiotemporal trident network as the data preprocessing layers. Specifically, for each input data unit, to reduce computations, the max-pooling layer is first used to reduce the size in the spatial domain, and then the residual noise signals are extracted by the SRM layer. The stride of the max-pooling layer in the temporal detection network is set to $1\times3\times3$, while it is set to $1\times2\times2$ in the spatial localization network. The SRM layer is a 3D convolutional layer with three specified parameter kernels used to output three different high-frequency residual signals of video frames. Fig. 2 shows that the three convolution kernels with different colors correspond to three types of high-frequency residual maps with different colors. The three convolution kernels of the SRM layer are shown in Fig. 3.

B. The prediction mechanisms based on the spatiotemporal trident network

For the input of the temporal detection algorithm, as shown in Fig. 4, we set the middle frame Frame$_t$ as the target frame in every five consecutive frames and set Frame$_{t-2}$, Frame$_{t-1}$, Frame$_{t+1}$, Frame$_{t+2}$ as the auxiliary frames. Then, we obtain the following ten input cases (we assume that the number of consecutive forged frames in a video is not fewer than five frames). We use rectangles with two different colors to represent the pristine frame and the forged frame. For each input case, 1) if three consecutive frames are pristine, their corresponding branch is represented by 0; 2) if three consecutive frames are forged, their corresponding branch is represented by 1; 3) if three consecutive frames contain both pristine frame and forged frame, the corresponding branch is represented by X. In this way, we can observe that if at least one of the three branches is 1, the middle frame must be forged, as shown in the dotted box at the bottom of Fig. 4; similarly, if at least one branch is 0, the middle frame must be pristine. Therefore, we can design a deep learning algorithm for learning this judging mechanism. In addition, the branch with pseudocode 1 can more reliably locate the tampered region through a localization algorithm. Thus, we can further design a spatial localization algorithm for the input Case 4∼8.

When X exists in the three branches, we call the input data unit, in this case, hybrid frames. Case 1 and Case 6 are the most input cases in the dataset. To strengthen the temporal detection algorithm's detection capability for the hybrid-frame cases, we set a frame weight called X_weight. When X exists in the three branches, we set X_weight to 1; otherwise, we set X_weight to 0, which is an extra penalty coefficient of the loss function for hybrid frames.

In fact, for the temporal detection algorithm, we do not intend to make the three branches output specific results such as 0, 1, or X, but output results as three vectors, and then use a BiLSTM to perform decoding. The three-time step of

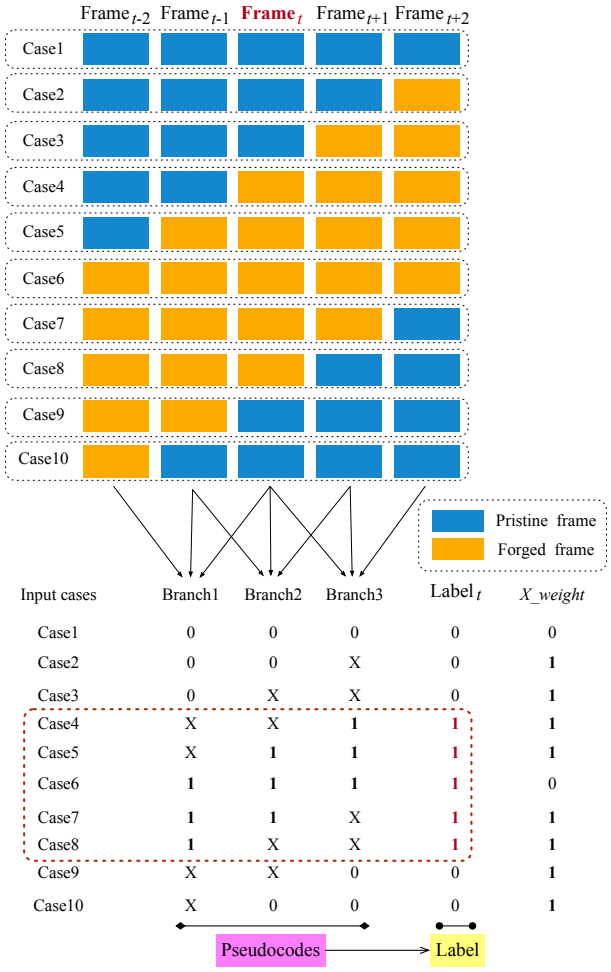


Fig. 4: Schematic diagram of the spatiotemporal trident network prediction mechanism. To aid our analysis of the potential relationship between data input and output, we use simple pseudocode to represent the input case of each branch.

BiLSTM is equivalent to three referees to decode and vote through the output vectors of three branches and finally turn it into a classification problem. The specific implementation of the temporal detection algorithm is given in Sec. IV.

For the spatial localization algorithm, we assume that the temporal detection algorithm has correctly classified the middle frames (whose input cases such as Case 4∼8 in Fig. 4) as forged frames, and it is assumed that the three branches in the spatial localization algorithm can all independently predict the tampered region within a forged frame. For the input Case 4∼8, since the number of consecutive forged frames in the input five frames is not less than 3, at least one 3D CNN branch of the spatial localization algorithm can predict reliably. The branch that can predict reliably will output a series of prediction regions with high confidences, while the branch that cannot predict reliably may provide some prediction regions with low confidences. Therefore, we perform nonmaximum suppression (NMS) operations on the three branches' output results and filter out a predicted box with the best confidence as the final prediction result of the

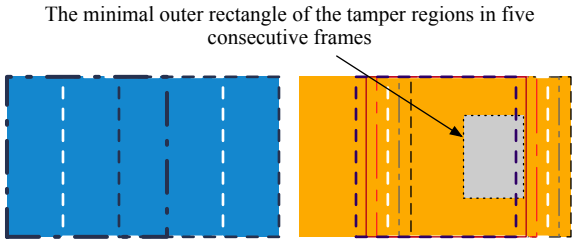


Fig. 5: Sampling methods of the pristine frames and the forged frames. The size of each positive and negative sample data generated by cropping is 5×(720×720)×3, where 5 represents the number of consecutive frames, and 3 represents the number of image channels.

spatial localization algorithm. The specific implementation of the spatial localization algorithm is provided in Sec. V.

IV. The Temporal Forgery Detection Algorithm

The temporal forgery detection algorithm aims to obtain the sequence of forged frames in the video by classifying each frame of the video as a pristine frame or a forged frame.

A. Dataset processing strategy for temporal forgery detection

Since the number of pristine frames in the video dataset is several times the number of forged frames, directly using the deep learning algorithm will affect the training effect due to the categories' imbalance. Therefore, we oversample the forged frames' tampered regions by cropping and undersample the pristine frame; then, an equal number of positive and negative samples are obtained. In this way, the scale of the sampled dataset can meet the deep learning algorithms' training requirements, which solves the problem that the small scale of the dataset is not suitable for deep learning algorithms mentioned by Aloraini et al. [2].

The cropping strategy [20] is shown in Fig. 5. For the resolution of our dataset of 1,280×720, the data of size 5×(720×720)×3 are sampled three times in five consecutive frames (whose middle frame is a pristine frame) by the same stride, while the data of size 5 ×(720×720)×3 are sampled thirteen times in five consecutive frames (whose middle frame is a forged frame) by smooth stride. It should be noted that every cropping operation is five frames aligned. The label of each cropped data unit is set to the middle frame label in the five frames, and then we obtain a large-scale and balanced dataset of positive and negative samples to suit our deep learning algorithms. The cropping method of the validation set is the same as that of the training set. Additionally, the testing set is cropped in the same way as the pristine frame, as shown in Fig. 5 (a). We divide the test input of each frame into three data units. If all three data units are predicted as pristine frames, the middle frame is detected as the pristine frame; if at least one of the three data units is predicted to be a forged frame, the middle frame is detected as a forged frame.

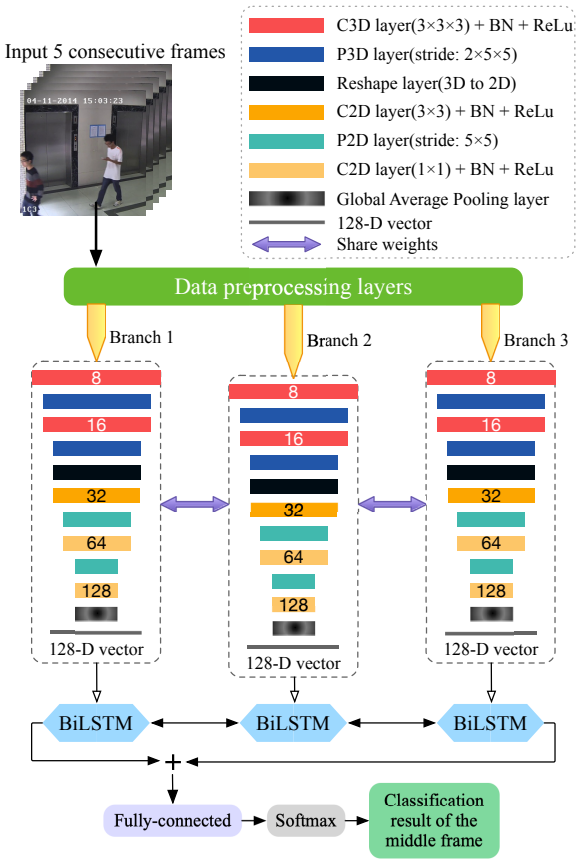


Fig. 6: Model structure of the temporal detection network. We use the 128-dimensional vectors to replace coding 0, 1, X in Fig. 4. We use a three-time-step BiLSTM to learn the classification mechanism we analyzed in Fig. 4 to classify the middle frame in five consecutive frames.

B. The structure of the temporal forgery detection network

The structure of the temporal forgery detection network is shown in Fig. 6. The input is the cropped data units with a size of 5×(720×720)×3. The input data first pass through the data preprocessing layers of the spatiotemporal trident network in Fig. 2, where it is sliced into three branches of data streams. Then, the three data streams enter the three 3D CNNs, where the three 3D CNNs share the weights to ensure uniform coding. Furthermore, the three 3D CNNs extract the high-frequency residual features of three consecutive frames in the spatiotemporal domain and encode them into three 128-dimensional vectors. We use BiLSTM as the decoder, and the output of the decoder is the sum of the output states in two directions of the BiLSTM. Finally, the fully connected layer and the softmax layer are used to convert the result into a binary classification problem to obtain the sequence of the forged frames in the forged video.

C. The encoder of the temporal forgery detection algorithm

The meaning of each layer of the encoder 3D CNN is shown in the dotted box in the upper right corner of Fig. 6. Different colors represent different operating layers, where C3D means 3D convolution, P3D means 3D average pooling; similarly,

C2D means 2D convolution, P2D means 2D average pooling. The number in the color block representing the convolutional layer indicates the number of convolution kernels. After each convolution layer, the batch normalization operations BN and ReLU are performed. The first two convolutional layers are C3D layers, and the size of the convolution kernel is 3×3×3. After each C3D layer, P3D is performed, and its stride is 2×5×5. The data with a temporal dimension of 3 are reduced to 1 after two P3D operations. The feature maps are transformed into 2D through the reshape layer and then pass through three C2D layers. After each C2D layer, P2D is performed, and the pooling stride is 5×5. The first C2D layer uses convolution kernels with a size of 3×3, and the next two C2D layers use convolution kernels with a size of 1×1 to increase the number of feature maps to 128. Finally, we use the global average pooling layer to transform the feature maps into a 128-dimensional vector.

D. The decoder of the temporal forgery detection algorithm

The decoder BiLSTM is composed of a forward LSTM [26] and a backward LSTM. LSTM can learn long-distance dependencies, while BiLSTM can better learn bidirectional semantic dependencies, both of which are commonly used to model contextual information in natural language processing tasks. In this paper, we use a BiLSTM as the decoder for learning the judging mechanism analyzed in Fig. 4. The BiLSTM performs decoding operations by analyzing the output vectors of three branches, and the decoded information is used for frame classification. The structure of LSTM is shown in Fig. 7. The inputs at time-step t include the following three items:

1) The hidden layer state h_{t-1} at the previous time-step;
2) The cell state C_{t-1} at the previous time-step;
3) The input x_t at the current time-step.

The following mathematical expressions express the detailed calculation process of the LSTM unit at time-step t:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{2}$$

$$\widetilde{C}_t = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{4}$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \tag{5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{6}$$

where f_t represents the value of the forget gate, i_t represents the value of the memory gate, $\widetilde{C}_t$ represents the temporary cell state, C_t represents the current unit state, o_t represents the value of the output gate, and h_t represents the hidden state layer.

The specific decoding strategy [27], [28] of BiLSTM in this paper is shown in Fig. 8. For the task of video frame classification, the decoding output scheme we used is $[h_{L2} + h_{R2}]$. The last hidden states of the forward LSTM and the backward LSTM are vector-added and used as the decoding output. In our experiment, we use BiLSTM with two layers stacked, and the number of neurons in both layers is set to 64.

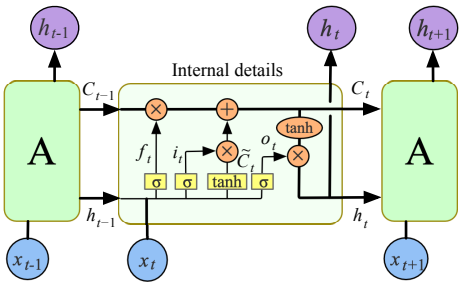


Fig. 7: The structure of LSTM [27].

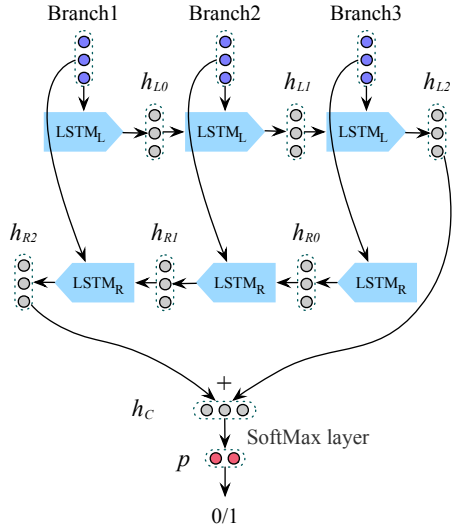


Fig. 8: Schematic diagram of the BiLSTM decoding strategy.

E. The loss function of the temporal forgery detection algorithm

The loss function of our temporal forgery detection algorithm is defined as follows:

$$\begin{aligned} Loss = &\frac{1}{N}\sum_{1}^{N} FL\left(p,\ y,\ \alpha,\ \gamma\right) \\ &+\frac{\beta}{N}\sum_{1}^{N}\left(X\_weight \cdot CE\left(p,\ y\right)\right) \end{aligned} \tag{7}$$

where N represents the training batch size, FL is the focal loss [29] function used for the classification task of most video frames, p is the prediction confidence of each frame, y is the label for each frame, α is a factor to adjust the imbalance of the categories, and γ is a factor to adjust the imbalance of hard and easy samples. CE is the cross-entropy function. X_weight is the hybrid-frame weight defined in Fig. 4, which is used for the case that the input data unit contains both the pristine frame and the forged frame; only in this case, the second part of loss

is not 0, and β is the weight of the extra penalty item. Since we have equalized the positive and negative samples, we set α to 0.5 and γ to 2 in our experiment. Furthermore, the focal loss function uses the following form defined in [29]:

$$FL\left(p_t\right) = -\alpha_t \left(1-p_t\right)^{\gamma} \log\left(p_t\right) \tag{8}$$

where p_t and α_t are defined in our method as follows:

$$p_t = \begin{cases} p, & y = 1 \\ 1-p, & y = 0 \end{cases} \tag{9}$$

$$\alpha_t = \begin{cases} \alpha, & y = 1 \\ 1-\alpha, & y = 0 \end{cases} \tag{10}$$

Therefore, the FL used in Equation 7 can be expressed in the following specific form:

$$FL\left(p,\ y,\ \alpha,\ \gamma\right) = \begin{cases} -\alpha\left(1-p\right)^{\gamma} \log\left(p\right), & y = 1 \\ -\left(1-\alpha\right) p^{\gamma} \log\left(1-p\right), & y = 0 \end{cases} \tag{11}$$

It is easy to see that after the model is trained for several epochs, if the model's predicted result of a sample is close to its label, then this sample will only contribute a small loss value, such a sample is an easy one for the model; on the contrary, a hard sample will contribute a far greater loss value than an easy one. Therefore, the larger the value of γ is, the more the model's training focus is biased towards hard samples. We set γ to 2 in all our experiments because the best performance was achieved when $\gamma = 2$ in [29].

V. The Spatial Forgery Localization Algorithm

The spatial forgery localization algorithm aims to further locate the tampered region within the video frame that has been judged to be a forged frame by the temporal forgery detection algorithm. Since the tampered area annotations in the dataset only labeled the approximate rectangular area without the pixel-level classification accuracy, we use RPN to predict the circumscribed rectangle of the tampered area. Otherwise, semantic segmentation in computer vision is a better choice.

A. Dataset processing strategy for spatial forgery localization

Since we need to locate the tampered region within the entire forged frame, the training dataset only needs consecutive forged frames without cropping. Therefore, the strategy of expanding the dataset no longer uses cropping operations but adopts horizontal flipping, vertical flipping, and horizontal-vertical flipping operations for the five consecutive forged frames, and the tampered region labeling should be transformed accordingly. Every five consecutive frames are packed into an input data unit of the spatial localization algorithm. Similar to the data labeling in the temporal forgery detection, the tampered region label of the middle frame in five frames is used as the label of the input data unit.

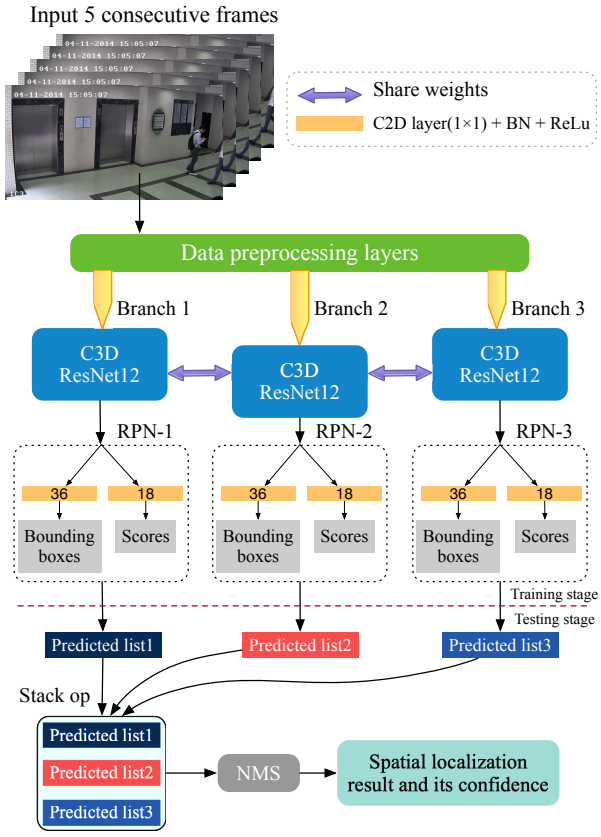


Fig. 9: Model structure of the spatial forgery localization network. The feature maps generated by C3D-ResNet12 are used for classification and regression tasks of bounding boxes in each branch.

B. The structure of the spatial localization network

The structure of the spatial localization network is shown in Fig. 9. The input is the flipped data units with a size of $5\times(720\times1280)\times3$, where 3 still represents the number of frame channels. The input data pass through the data preprocessing layers of the spatiotemporal trident network in Fig. 2, and the data stream is divided into three branches by the slicing operation. Analogous to the temporal forgery detection network, each branch of data streams enters a backbone network named C3D-ResNet12 in Fig. 10, and the three backbone networks share the weights to ensure uniform coding. At the end of each backbone network, the 3D feature maps from different times are stacked and reshaped to 2D. The outputs of 2D feature maps in three branches are used as raw materials for three independent RPNs to predict three localization lists of the tampered region.

C. The structure of our C3D-ResNet12

We designed a backbone network named C3D-ResNet12, with 12 layers of the 3D residual network, as shown in Fig. 10. We use different color blocks to represent different operations, as shown in the dotted box at the top of Fig. 10, where the number in the color block representing the convolutional layer indicates the number of convolution kernels. Our backbone

network consists of three blocks and a feature map slicing operation. Each block has a shortcut layer from the first layer to the last layer; adjacent blocks also have a shortcut layer from the third layer of the previous block to the second layer of the next block. Both the shortcut layer and the layer to be connected provide half the number of feature maps and then use ReLU after concatenation. The C3D kernel size of all shortcut layers in the backbone network is 1×1×1, while the size of other C3D kernels is 3×3×3. In each block, the first three C3D layers use dilated convolutions with a dilation rate of 5, and the fourth C3D layer uses a 1×2×2 convolution stride to replace the pooling layer. As shown in Fig. 10, the input data size of the backbone network is 3×(360×640)×3, where the first number 3 represents the time dimension of three consecutive frames, the last number 3 represents the number of image channels, and the output size of Block 3 is 3×(45×80)×32, where 3 still represents the time dimension, 32 represents the number of feature maps, and 45×80 represents the size of feature maps. Here, we slice the feature maps with a time dimension of 3, stack the feature maps from different times and reshape them to 2D feature maps, and finally output the feature map with a size of (45×80)×96. We use the output feature maps as the input of the RPN.

D. The RPN in the spatial forgery localization

For the RPN algorithm, we use C2D layers with a kernel size of 1×1 to train the classification and regression of bounding boxes and output the coding sequence of prediction boxes and their classification confidences. To reduce the false detection rate of tampered regions, we set the ratio of the number of foreground training boxes (tampered regions) fg_num to background training boxes (original regions) bg_num in each frame to 1:λ, and the constraint formulas [10] are as follows:

$$fg\_num = \min\left(fg\_sum,\ \frac{roi\_num}{\lambda+1})\right) \tag{12}$$

$$bg\_num = \min\left(roi\_num - fg\_num,\ fg\_num \times \lambda\right) \tag{13}$$

where fg_sum is the total number of foreground boxes and roi_num is a hyperparameter that controls the density of training boxes. In our experiment, we set roi_num to 128 and λ to 5.

The following is the initialization strategy of the sizes of anchor boxes in the RPN algorithm: for the widths and heights of all tampered regions in the training set, we use the K-means clustering algorithm to iterate out three width values and three height values. Then, we obtain nine pairs of width-height values to initialize our anchor boxes.

E. The loss function of the spatial forgery localization algorithm

The following loss function is the sum of the costs of classification and regression in three branches, and it is defined as:

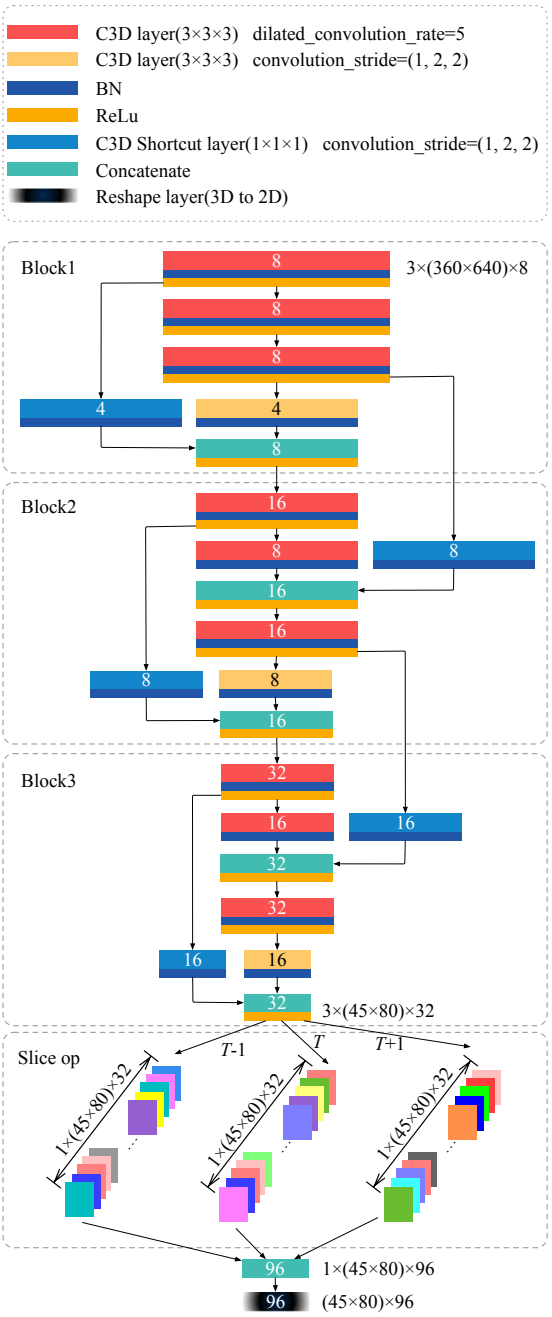


Fig. 10: The structure of C3D-ResNet12. The specific parameters and details of each layer can be inferred from the diagram.

$$Loss = \sum_{i=1}^{3}\left(\frac{1}{N}FL\left(preds_i,\ labels,\ \alpha,\ \gamma\right) + \frac{1}{M}GL\left(decode\left(bbox\_preds_i\right),\ decode\left(bbox\_targs\right)\right)\right) \tag{14}$$

where FL is the focal loss function, which is used to classify each training box within a single frame as a foreground box (tampered region) or a background box (original region). N represents the number of training boxes. The parameter $preds$ is the sequence of predicted classifications of the training

boxes. The parameter $labels$ is the sequence of the training boxes' classification labels, where $labels$ is defined as:

$$labels = \begin{cases} 1, \ IoU\left(box_{anchor}, \ box_{ground\_truth}\right) \geq 0.7 \\ 0, \ IoU\left(box_{anchor}, \ box_{ground\_truth}\right) \leq 0.3 \\ ignore; \ otherwise, \end{cases} \tag{15}$$

We use α to adjust the imbalance of foreground boxes and background boxes involved in training and use γ to control the balance of hard and easy samples. M represents the number of foreground boxes involved in training. GL (GIoU_loss) [30] is a loss function used for bounding box regression, whose parameters are the predicted region coordinates and the labeling coordinates. Therefore, the coordinates encoded with the anchor boxes should be decoded. The GIoU_loss algorithm defined in [30] is as follows:

Algorithm: Loss of generalized intersection over union

input : Two arbitrary convex shapes: $A, B \subseteq \mathbb{S} \in \mathbb{R}^n$

output : IoU, $GIoU$, $GIoU\_loss$

1 For A and B, find the smallest enclosing convex object C, where $C \subseteq \mathbb{S} \in \mathbb{R}^n$

2 $IoU = \frac{|A \cap B|}{|A \cup B|}$

3 $GIoU = IoU - \frac{|C \backslash (A \cup B)|}{|C|}$

4 $GIoU\_loss = 1 - GIoU$

F. Prediction strategy of the spatial localization algorithm

In the training stage, we train the three branches of the spatial locator synchronously to make each branch have the ability of independent prediction. In the testing stage, as shown in Fig. 9, we stack the three branches' prediction lists into a complete prediction list (including the coordinates and their confidences of the prediction region). Finally, we use NMS to obtain the predicted box with the highest confidence as the final localization box.

VI. Experimental analysis

A. Datasets

To date, there are few publicly available datasets of video tampering; the three well-known datasets of video content tampering are SULFA [31], REWIND [32] and SYSU-OBJFORG [1]. There are only five videos in SULFA that belong to the type of video object removal tampering, and each video is approximately 10 seconds in length, with a resolution of 320 × 240 and a frame rate of 30 FPS. REWIND is based on SULFA, including ten original videos and ten copy-pasted forged videos, with the same frame rate and resolution as SULFA. The amount of data available for the above two datasets is too small, which is not conducive to deep learning methods for training. To the best of our knowledge, SYSU-OBJFORG is the most massive dataset of video object removal tampering reported by Chen et al. [1], including 100 original videos and 100 forged videos corresponding to the original videos, all of which are from static commercial surveillance cameras, which are 3 Mbits/s. Each video's length is approximately 11 seconds, with a resolution of 1,280 × 720 (720p) and a frame rate of 25 FPS. All videos are compressed in the H.264/MPEG-4 encoding format. Each forged video contains one or two forged segments lasting from 1 to 5 s. The quality of tampering is relatively high, so the trace of tampering is completely invisible to the naked eye. All forged video clips are recompressed with the same parameters as in the corresponding pristine video clips. After data sampling, the quantity of data can meet our deep learning algorithms; thus, we use SYSU-OBJFORG to carry out all the work. To make most of the information of each frame, it should be noted that we decompressed all videos into the BMP image format for processing.

B. Experimental settings

The spatiotemporal localization algorithms based on deep learning in this paper are implemented based on the TensorFlow framework, run on the Ubuntu system, and use the NVIDIA GeForce GTX1080ti GPU. We use Adam as the optimizer and set the learning rate to 0.001, the $momentum$ to 0.9, the $l2$ regularization parameter to 0.0005, and the parameter initialization standard deviation to 0.1.

The details of the dataset division and the performance of the models are shown in Table I. Note that the original video divided in Table I(a) and its corresponding forged video are divided in pairs for our temporal forgery detection algorithm. Based on the above experimental settings, we show the test speed performance of our two models in Table I(b).

C. Training and testing of temporal forgery detection

The batch size of the temporal forgery detection algorithm in the training stage is set to 64 (i.e., the dimension of the image block fed into the neural network each time is 64×5×(720×720)×3). In the testing stage, the batch size is set to 3 (i.e., the dimension of the image block fed into the neural network each time is 3×5×(720×720)×3), and the three batches of test data are generated by the cropping operations in Fig. 5 (a); therefore, each frame's classification result is determined by the classification results of the three data pieces. The classification strategy is as follows: if all three data pieces are predicted as the pristine frame, then the middle frame is detected as a pristine frame; otherwise, it is detected as a forged frame.

For the test evaluation metrics for temporal forgery detection, we still use the following six evaluation metrics defined by Chen et al. [1]:

$$PFACC = \frac{\sum correctly\_classified\_pristine\_frames}{\sum pristine\_frames} \tag{16}$$

$$FFACC = \frac{\sum correctly\_classified\_forged\_frames}{\sum forged\_frames} \tag{17}$$

$$FACC = \frac{\sum correctly\_classified\_frames}{\sum all\_the\_frames} \tag{18}$$

$$Precision = \frac{T_P}{T_P + F_P} \tag{19}$$

$$Recall = \frac{T_P}{T_P + F_N} \tag{20}$$

TABLE I: DETAILS OF OUR EXPERIMENTS. (a) THE DETAILS OF THE DIVISION OF THE VIDEO DATASET AND THE NUMBER OF SAMPLES GENERATED BY THE DIVIDED VIDEOS. (b) THE PARAMETER SCALE OF THE MODELS AND THEIR TRAINING AND TESTING PERFORMANCE.

Our algorithms	Training set		Validation set		Test set	
	Number of videos	Number of samples	Number of videos	Number of samples	Number of videos	Number of samples
The temporal forgery detection algorithm	60 (original) + 60 (forged)	170074	20 (original) + 20 (forged)	57174	20 (original) + 20 (forged)	34488
The spatial forgery localization algorithm	60 (forged)	19972	/	/	40 (forged)	4580

(a)

Our algorithms	Total number of trainable parameters	Time required for model training	Speed of model testing
The temporal forgery detection algorithm	184442	5 $\sim$ 7 days	27 FPS
The spatial forgery localization algorithm	114780	1 $\sim$ 2 days	5 FPS

(b)

TABLE II: COMPARISON OF TEMPORAL FORGERY DETECTION RESULTS OF DIFFERENT ALGORITHMS.

Approaches \Evaluation metrics (%)	*PFACC*	*FFACC*	*FACC*	*Precision*	*Recall*	*F1Score*	*IoU*	*HFACC*
CC-PEV [1]	99.90	83.94	95.71	90.48	91.80	91.13	—	—
SPAM [1]	99.71	76.86	92.47	78.90	83.04	80.92	—	—
CF* [1]	99.50	77.55	94.15	87.06	85.87	86.46	—	—
CDF [1]	99.96	84.07	95.88	90.20	91.01	90.60	—	—
SRM [1]	99.92	76.40	93.70	83.10	82.68	82.89	—	—
CC-JRM [1]	99.96	84.39	96.59	93.15	91.51	92.32	—	—
J + SRM [1]	**99.99**	84.90	96.59	92.80	91.58	92.18	—	—
F-DIFF + HPF + CNN [20]	98.82	91.05	96.90	98.12	92.10	94.38	—	—
Temporal CNN [21]	—	96.04	97.49	—	—	—	—	—
TF-SA [18]	—	—	—	93.30	93.30	93.30	—	—
S-PA [2]	—	—	—	95.51	94.44	94.97	90.43	—
SRM + C3D (single branch)	98.99	97.96	98.78	96.28	97.96	97.11	94.40	75.67
Our approach (β=0)	99.21	97.88	98.93	97.07	97.88	97.47	95.08	91.21
Our approach (β=1)	99.40	97.55	99.01	97.75	97.55	97.65	95.41	89.86
Our approach (β=2)	99.50	**98.75**	**99.34**	**98.14**	**98.75**	**98.44**	**96.94**	**93.24**
Our approach (β=5)	99.43	98.63	99.26	97.89	98.63	98.26	96.58	**93.24**
Our approach (β=10)	99.30	98.54	99.14	97.41	98.54	97.97	96.03	92.56

$$F1Score = \frac{2T_P}{2T_P + F_P + F_N} \tag{21}$$

where $PFACC$ is the pristine frame accuracy, $FFACC$ is the forged frame accuracy, and $FACC$ is the accuracy of all frames. $Precision$, $Recall$, and $F1score$ can be calculated by three indicators: T_P (representing the number of forged frames that are correctly predicted), F_P (representing the number of pristine frames that are incorrectly predicted as forged ones) and F_N (representing the number of forged frames that are incorrectly predicted as pristine ones). It should be noted that $FFACC$ and $Recall$ are completely equal in this paper, while in [1], these are two completely different metrics because their calculations are based on video clips, not on frames.

To compare with the method in [2], we also calculated the intersection over union (IoU) between the detected result and the ground truth in the temporal forgery detection:

$$IoU = \frac{T_P}{T_P + F_P + F_N} \tag{22}$$

To verify the detection accuracy of the temporal forgery detection algorithm when the input cases are hybrid frames (e.g., Cases 2$\sim$5 and 7$\sim$10 in Fig. 4), we define the detection accuracy of hybrid frames as follows:

$$HFACC = \frac{\sum correctly\_classified\_hybrid\_frames}{\sum all\_the\_hybrid\_frames} \tag{23}$$

We validated our method on the SYSU-OBJFORG dataset. Among 100 pairs of videos, we randomly divided the training set, validation set, and testing set, as shown in Table I(a). According to the different values of β in the loss function, multiple experiments were carried out, and the test results of temporal forgery detection are shown in Table II. Comparing the results of SRM + C3D (single branch) and F-DIFF + HPF + CNN, the FACC and recall values have been greatly improved. The largest difference between them is that SRM + C3D (single branch) uses 3D CNN, which shows that 3D CNN is more suitable for processing video data than 2D CNN because 3D CNN can extract the tampering traces in the video more sufficiently. Comparing SRM + C3D (single branch) and our approach ($\beta = 0$), there is a significant improvement in almost all metric values, especially HFACC, which shows that the proposed spatiotemporal trident network is effective and that the prediction ability of hybrid-frame cases has been greatly improved. When $\beta = 0$, the loss function is equal to the focal loss function. When $\beta > 0$, the loss function has

TABLE III: THE TEST RESULTS OF THE SPATIAL FORGERY LOCALIZATION ALGORITHM.

Approaches	Number of test frames	$Rate_{suc}$ (%)	IoU_{mean} (%)
VGG16 + RPN (single branch)	4557	68.61	40.88
SRM + VGG16 + RPN (single branch)	4557	89.99	45.46
3D-CNN + RPN (single branch)	4594	46.52	29.08
SRM + 3D-CNN +RPN (single branch)	4594	94.47	49.07
Our approach (α=0.5)	4580	95.08	61.65
Our approach (α=0.6)	4580	95.74	61.76
Our approach (α=0.7)	4580	96.06	62.07
Our approach (α=0.75)	4580	96.06	62.03
Our approach (α=0.8)	4580	95.93	**64.58**
Our approach (α=0.85)	4580	**96.85**	63.21.

TABLE IV: TEN RANDOM EXAMPLES OF THE COMPLETE TEST RESULTS OF OUR SPATIOTEMPORAL FORGERY LOCALIZATION ALGORITHMS.

Test videos (video index in the dataset)	Stage of temporal domain detection				Stage of spatial domain localization	
	Number of video frames	Number of forged frames	The correctly predicted number of forged frames	$FACC$ (%)	$Rate_{suc}$ (%)	IOU_{mean} (%)
Video 1 (100)	291	180	180	97.59	97.77	65.48
Video 2 (093)	292	112	111	97.26	93.75	63.66
Video 3 (053)	292	81	81	99.31	100.00	57.22
Video 4 (019)	284	139	139	97.18	95.68	59.26
Video 5 (072)	292	130	128	98.63	100.00	66.33
Video 6 (009)	284	99	96	97.53	90.90	39.61
Video 7 (040)	284	88	88	100.00	100.00	72.26
Video 8 (006)	285	123	123	99.64	100.00	70.48
Video 9 (008)	284	140	140	100.00	93.57	64.34
Video 10 (033)	285	136	136	99.29	99.26	75.85

an extra penalty item for the hybrid-frame cases, so that the model focuses more on the training of the hybrid-frame. When β takes different values, we found that the performance is the best when $\beta = 2$. It is worth noting that although we have taken a series of measures to improve the test accuracy of the hybrid-frame cases, HFACC is still much lower than other metric values. A reasonable explanation is that the value of the 128-dimensional vector obtained by the branch represented by X in Fig. 4 may be random and meaningless, which may have some misleading effect on BiLSTM's prediction.

D. Training and testing of the spatial forgery localization

In the spatial localization experiment, the training set's video allocation lists are the same as the temporal forgery detection, while the difference is that the training set of the spatial localization algorithm only requires consecutive forged frames. We process the training set according to Sec. V-A's method, and the testing set does not need data enhancement operations, such as flipping or cropping. The batch size of the spatial localization algorithm in the training stage is set to 2 (i.e., the dimension of the image block fed into the neural network each time is $2\times5\times(720\times1280)\times3$). In the testing stage, the batch size is set to 1 (i.e., the dimension of the image block fed into the neural network each time is $1\times5\times(720\times1280)\times3$). Furthermore, we take the predicted box with the highest confidence among the predicted lists of three branches as the final localization box.

If the intersection over union of the final localization box and the real tampered region is 0 or the confidence is less than 0.8, it is defined as a missed localization frame F_{mis}. Otherwise, it is defined as a successful localization frame F_{suc}. We define the successful localization rate $Rate_{suc}$ as:

$$Rate_{suc} = \frac{\sum F_{suc}}{\sum F_{suc} + \sum F_{mis}} \tag{24}$$

The mean intersection over union (IOU_{mean}) of the predicted boxes and the tampered regions in all the test forged frames is defined as:

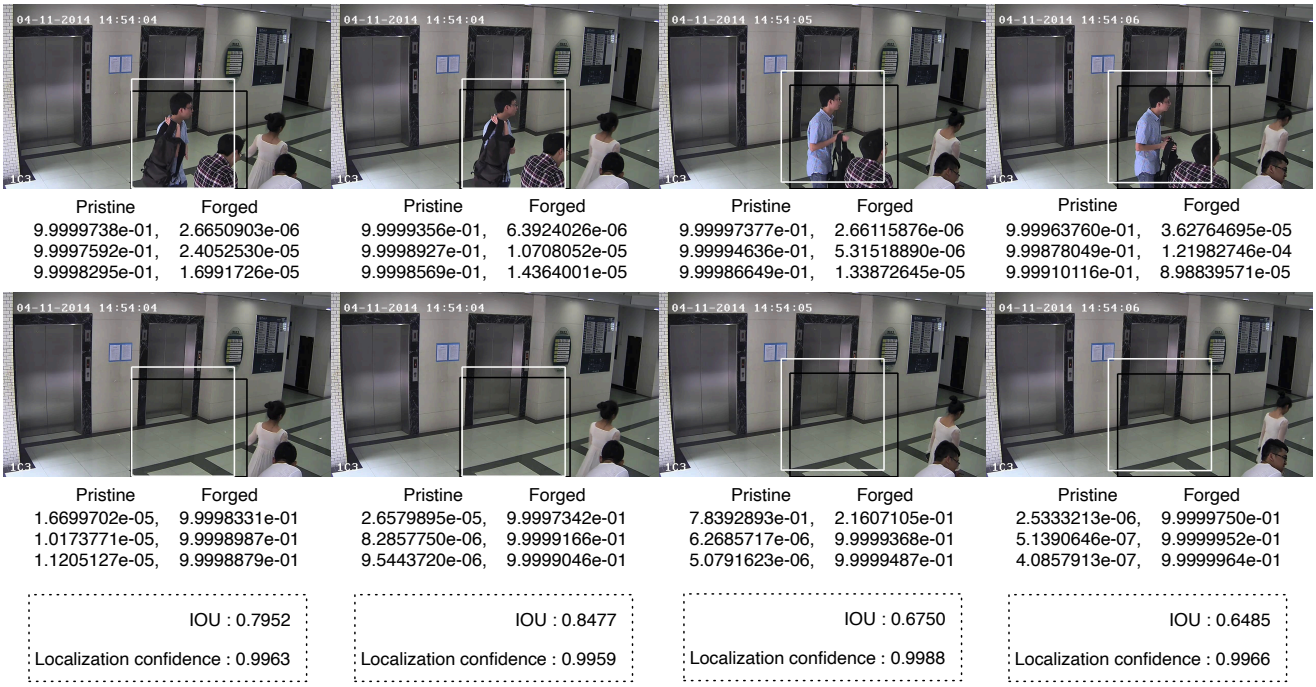
$$IoU_{mean} = \frac{1}{N_{suc}} \sum_{i} IoU_i \tag{25}$$

where N_{suc} represents the total number of successful localization frames, and i represents the subscript of the successful localization frames.

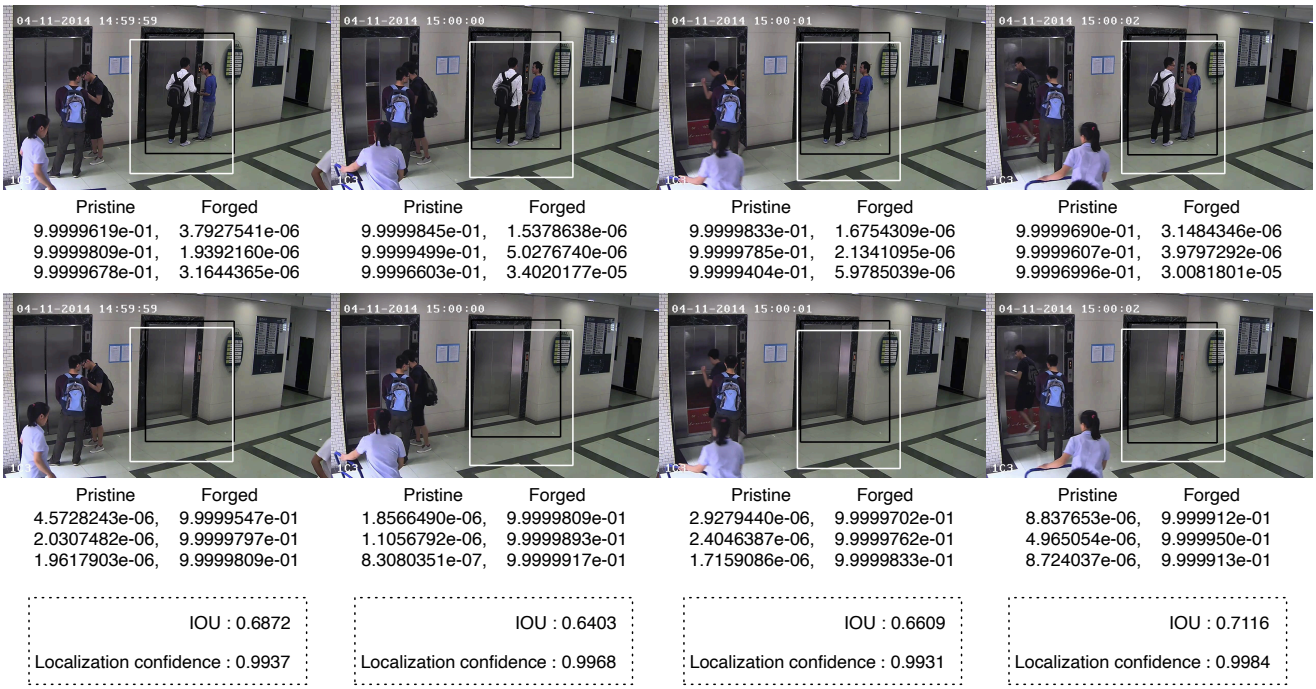
Table III shows the results of the spatial localization experiments, and Fig. 11 shows two examples of spatial forgery localization effect maps in two videos. From the comparison results of the single-branch methods in Table III, we can see the obvious effect of the SRM layer and C3D layer. In a series of experiments with different α values, we find that when the α value nearly balances the positive and negative samples, our approach performs best and is greatly improved compared to the single-branch methods. Below each frame in Fig. 11, the test results of the temporal forgery detection algorithm and the spatial forgery localization algorithm are given. The test result of the temporal forgery detection algorithm gives the prediction confidences of the three test samples generated by each frame, and the test result of the spatial forgery localization algorithm gives the IOU of the prediction box and the label box and the confidence of the prediction box.

E. Complete test of spatiotemporal forgery localization

We randomly selected ten forged videos from the testing set for complete temporal and spatial localization tests. First, we



(a)



(b)

Fig. 11: Two examples of the spatial forgery localization results of the tampered regions. In each example, the upper row gives the frames of an original video, while the lower row gives the frames of the forged video corresponding to the original one. In each forged frame, the black box is the real tampering area (ground truth), and the white box is our localization result. To facilitate comparison, we also made corresponding marks on the original frames.

perform temporal forgery detection on test video frames and record the forged frames detected by the temporal detection algorithm; then, we input these forged frames in the form required by the spatial localization algorithm to locate and evaluate the spatial localization results. The test results are shown in Table IV.

VII. Conclusion

In this paper, we proposed a spatiotemporal trident network based on SRM and C3D, and we analyzed the theoretical basis

of our spatiotemporal trident network to improve the accuracy of spatiotemporal localization algorithms. According to the theoretical basis, we designed a temporal forgery detection algorithm for forged frames and a spatial forgery localization algorithm for the tampered region within each forged frame. In the structure of the temporal forgery detection network, 3D CNN is used as the encoder, and BiLSTM is employed as the decoder, which improves the accuracy of temporal forgery detection while also significantly improving the localization accuracy in the hybrid-frame input cases. In the structure of the spatial localization network, C3D-ResNet12 was designed to stack the feature maps in the temporal domain, and all the temporal feature maps were used as the basis for bounding box regression. Moreover, the three branches have independent spatial localization capabilities, which can significantly improve the spatial localization accuracy.

We solved two problems for the hybrid-frame input cases. (1) The temporal detection algorithm based on the spatiotemporal trident network solved the problem that single-branch 3D CNN cannot locate forged frames to every single frame. (2) The spatial localization algorithm based on the spatiotemporal trident network solved the problem of the single-branch structure's inaccurate spatial localization. Compared with traditional algorithms, our algorithms have significantly improved localization accuracy, but the computations is relatively large, requires GPU support, requires a large quantity of training data, and usually requires a long training period. How to simplify the model, reduce the amount of computation, and ensure accuracy at the same time will be the main work in the future.

REFERENCES

[1] S. Chen, S. Tan, B. Li, and J. Huang, "Automatic detection of object-based forgery in advanced video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 11, pp. 2138–2151, 2016.

[2] M. Aloraini, M. Sharifzadeh, and D. Schonfeld, "Sequential and patch analyses for object removal video forgery detection and localization," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2020.

[3] R. Chen, G. Yang, and N. Zhu, "Detection of object-based manipulation by the statistical features of object contour," *Forensic Science International*, vol. 236, pp. 164–169, 2014.

[4] X. Liao, Y. Yu, B. Li, Z. Li, and Z. Qin, "A new payload partition strategy in color image steganography," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 685–696, 2020.

[5] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Learning rich features for image manipulation detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1053–1061.

[6] P. Zhang, X. Wang, W. Zhang, and J. Chen, "Learning spatial–spectral–temporal eeg features with recurrent 3d convolutional neural networks for cross-task mental workload assessment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 1, pp. 31–42, 2019.

[7] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4207–4215.

[8] A. Aziz Sharfuddin, M. Nafis Tihami, and M. Saiful Islam, "A deep recurrent neural network with bilstm model for sentiment classification," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018, pp. 1–4.

[9] R. Ma, S. Teragawa, and Z. Fu, "Text sentiment classification based on improved bilstm-cnn," in *2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, 2020, pp. 1–4.

[10] S. Huang and D. Ramanan, "Expecting the unexpected: Training detectors for unusual pedestrians with adversarial imposters," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4664–4673.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[12] T. Uemura, J. J. Näppi, T. Hironaka, H. Kim, and H. Yoshida, "Comparative performance of 3D-DenseNet, 3D-ResNet, and 3D-VGG models in polyp detection for CT colonography," in *Medical Imaging 2020: Computer-Aided Diagnosis*, vol. 11314. SPIE, 2020, pp. 736 – 741.

[13] M. Saddique, K. Asghar, U. I. Bajwa, M. Hussain, and Z. Habib, "Spatial video forgery detection and localization using texture analysis of consecutive frames," *Advances in Electrical and Computer Engineering*, vol. 19, no. 3, pp. 97–108, 2019.

[14] S. Tan, S. Chen, and B. Li, "Gop based automatic detection of object-based forgery in advanced video," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2015, pp. 719–722.

[15] K. Sitara and B. Mehtre, "Differentiating synthetic and optical zooming for passive video forgery detection: An anti-forensic perspective," *Digital Investigation*, vol. 30, pp. 1 – 11, 2019.

[16] Y. Liu, T. Huang, and Y. Liu, "A novel video forgery detection algorithm for blue screen compositing based on 3-stage foreground analysis and tracking," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 7405–7427, 2018.

[17] L. D'Amiano, D. Cozzolino, G. Poggi, and L. Verdoliva, "A patchmatch-based dense-field algorithm for video copy–move detection and localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 3, pp. 669–682, 2019.

[18] M. Aloraini, M. Sharifzadeh, C. Agarwal, and D. Schonfeld, "Statistical sequential analysis for object-based video forgery detection," *electronic imaging*, vol. 2019, no. 5, pp. 543–1–543–7, 2019.

[19] J.-L. Zhong, C.-M. Pun, and Y.-F. Gan, "Dense moment feature index and best match algorithms for video copy-move forgery detection," *Information Sciences*, vol. 537, pp. 184–202, 2020.

[20] Y. Yao, Y. Shi, S. Weng, and B. Guan, "Deep learning for detection of object-based forgery in advanced video," *Symmetry*, vol. 10, no. 1, p. 3, 2017.

[21] A. Kohli, A. Gupta, and D. Singhal, "Cnn based localisation of forged region in object-based forgery for hd videos," *IET Image Processing*, vol. 14, no. 5, pp. 947–958, 2020.

[22] C. Feng, Z. Xu, S. Jia, W. Zhang, and Y. Xu, "Motion-adaptive frame deletion detection for digital video forensics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2543–2554, 2017.

[23] C. Cheng, P. Lv, and B. Su, "Spatiotemporal pyramid pooling in 3d convolutional neural networks for action recognition," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 3468–3472.

[24] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[25] J. Wei, H. Wang, Y. Yi, Q. Li, and D. Huang, "P3d-ctn: Pseudo-3d convolutional tube network for spatio-temporal action detection in videos," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 300–304.

[26] S. D. Kumar and D. Subha, "Prediction of depression from eeg signal using long short term memory(lstm)," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 1248–1253.

[27] X. Yin and C. Hang, "Arrhythmia classification based on cnn bilstm," in *2019 6th International Conference on Systems and Informatics (ICSAI)*, 2019, pp. 1105–1109.

[28] L. Wang, X. Bai, and F. Zhou, "Few-shot sar atr based on conv-bilstm prototypical networks," in *2019 6th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, 2019, pp. 1–5.

[29] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[30] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658–666.

[31] G. Qadir, S. Yahaya, and A. T. S. Ho, "Surrey university library for forensic analysis (sulfa) of video content," in *IET Conference on Image Processing (IPR 2012)*, 2012, pp. 1–6.
[32] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Local tampering detection in video sequences," in *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, 2013, pp. 488–493.

Quanxin Yang received his B.S. degree in internet of things engineering from Xuchang University, Henan, China, in 2017. He is now pursuing his Ph.D. in computer science and technology at Hangzhou Dianzi University, Hangzhou, Zhejiang, China. His current research interests include image processing, multimedia forensics, and big data.

Dongjin Yu is currently a Professor at Hangzhou Dianzi University, China. His current research focuses on data engineering, service computing and intelligent software engineering. He is the director of Institute of Big Data (IBD) and Institute of Computer Software (ICS) of Hangzhou Dianzi University. He is the senior members of IEEE and China Computer Federation (CCF). He is also a member of Technical Committee of Software Engineering CCF (TCSE CCF) and a member of Technical Committee of Service Computing CCF (TCSC CCF).

Zhuxi Zhang received the B.S. degree from Hangzhou Dianzi University, Hangzhou, Zhejiang, China, in 2020. She is currently pursuing the master's degree with the School of Cyberspace, Hangzhou Dianzi University. Her current research interests include information security, video forensics, deep learning, and computer vision.

Ye Yao received the M.S. degree in computer science and the Ph.D. degree in communication and information system from Wuhan University, Wuhan, China, in 2005 and 2008, respectively. He was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, from December 2016 to December 2017. He is currently an Associate Professor with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou. His research interests include multimedia forensics and information security.

Linqiang Chen received the B.S. degree in computational mathematics and software applications from Hangzhou University, Hangzhou, China, in 1983, and the M.S. degree in computer graphics from Zhejiang University, Hangzhou, in 1989. He is currently a Professor with the School of Cyberspace, Hangzhou Dianzi University, Hangzhou. His research interests include computer graphics and multimedia information processing.