Advances in Computer, Signals and Systems (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2024.080305
ISSN 2371-8838 Vol. 8 Num. 3

Video matting tampering detection based on time and space domain traces

Wenyi Zhu[1,a,*], Yulin Zhao[1,b], Yingqian Deng[1,c]

[1]School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, China

[a]zhuwenyi@mail.hnust.edu.cn, [b]zhaoyulin@mail.hnust.edu.cn, [c]dengyingqian@mail.hnust.edu.cn

[*]Corresponding author

Keywords: Video forensics, tamper region localisation, optical flow, frame residuals

Abstract: Deep learning-based videos usually leave imperceptible traces when tampered with. Tampered videos may be used for malicious video manipulation, which raises privacy and security concerns. Therefore the detection and localisation of tampered video traces is necessary. In this paper, we locate the tampering region by using the traces left behind by time and space domain information, and use VOS as a refinement network to improve the model performance. Firstly, the base network enhances the tampering traces by intra- and inter-frame residuals, and a dual-stream network is designed as an encoder to extract the special diagnosis from the frame residuals. Afterwards, a bidirectional convolutional LSTM and transposed convolution are embedded in the decoder to generate a prediction mask. Afterwards, a VOS network is used to obtain more accurate object boundaries. Extensive experimental results on public and synthetic manipulated datasets show that the proposed method can accurately locate tampered regions and outperforms and is robust to state-of-the-art methods.

1. Introduction

In today's digital age, video has become one of the widely used media for information dissemination. Video matting, as an important technique, allows users to extract, edit and recombine objects from videos to create new visual effects. With the continuous development of deep learning techniques, video matting has made great progress. Deep learning models can automatically learn and understand the objects in the video to achieve more accurate and efficient matting results. By combining deep learning techniques and video matting, researchers have developed many advanced algorithms and tools that open up new possibilities in video editing, virtual reality, special effects production, and more. This convergence provides strong support for the innovation and development of digital media and brings a broad prospect for future research and applications. However, at the same time, the phenomenon of tampered videos[1-3] has become more and more common, and we can see a large number of tampered videos on various media platforms and social networks. This phenomenon poses serious challenges to the authenticity and integrity of videos, making it difficult to determine the source, content and authenticity of videos.

Although there have been several studies focusing on image and video tampering detection, as a relatively new research area[1], passive detection of video matting still faces many techniques and challenges. In the literature, there are many works on passive forensics of digital images[2,3]. However, the research on passive forensics for digital video is still in its infancy. Traditional video matting tampering detection techniques usually process the temporal information of the video[4-6] in a more limited way, mainly focusing on the analysis of single-frame images, and lack the full use of temporal information such as object motion and temporal relationships in the video sequence, resulting in the possibility of overlooking the dynamic changes in the video tampering behaviour. With the development of deep learning, deep learning models can make better use of the temporal information in video sequences by continuously analysing and modelling the video frames, to more accurately detect tampering behaviours in videos, including object motion, dynamic changes, etc. However, there are only a few methods that try to deal with video matting tampering detection[7,8] and are underutilised for tampering edge traces. On the other hand, several image forensics methods can locate tampered video regions frame by frame[9], but they do not take advantage of the temporal correlation between video frames and perform poorly.

To address the above issues, in this paper, an end-to-end framework is proposed to locate regions tampered with by deep video. The contributions of this paper are summarised in three areas:

(1) This paper proposes a dual-stream network to learn features from residuals, and to mine the tampering leftover information based on time and space domains, intra-frame residuals and inter-frame residuals are used, where the intra-frame residuals are guided using optical flow to better get the tampering traces.

(2) Using the VOS model as an auxiliary task, the boundary traces of tampered objects are blurred and masked poorly, using VOS to obtain a more precise positioning of object boundaries.

(3) The results in the publicly available dataset and the self-established dataset show that the model in this paper has a clear advantage over the advanced manipulation detection models specifically.

2. Proposed Method

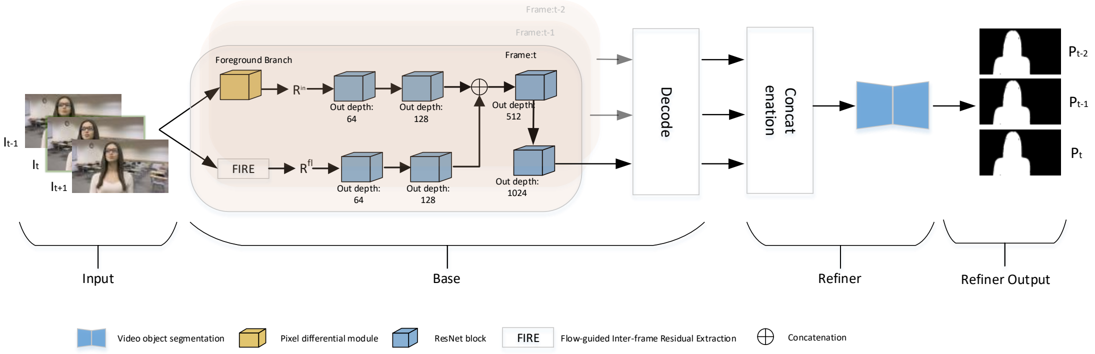

Figure 1: The overall framework of the proposed method.

The model consists of a base network Gbase and a refinement network Grefine as shown in Fig 1. Where the base network Gbase designs a two-stream network as an encoder that learns features from intra-frame residuals and inter-frame residuals and then encodes these two residuals separately for feature fusion, and a decoder network embedded with bi-directional convolutional LSTMs and transposed convolutions to model the correlation between consecutive frames as well as pixel prediction for each frame. The refinement network is composed of the VOS model, which is used as an auxiliary task to help the base network, a Fully Computational Network (FCN) with dense prediction is used, although the base network operates on each frame of the video, the boundary traces of the objects are blurred and the masking is not effective, the use of the VOS obtains a more precise positioning of the object boundaries without artefacts.

2.1. Intra-frame residual stream

The content of the tampered region is based on the known pixels from the same frame by capturing the boundary image boundary traces to locate the tampered region, the image extracted from adjacent frames will inevitably be left behind based on the spatial and temporal information for the extraction of intra-frame residuals, and the difference between the tampered and untampered regions is analysed through the experiments and is more pronounced in the high pass pre-filtered residuals. Therefore a pre-filtering module initialised with a high pass filter is designed to extract the image residuals to enhance the tampering traces. The high pass pre-filter module can highlight the edge information in the image or video, making the boundary between the tampered region and the original content more clear, and video tampering usually involves modification or replacement of pixel values which usually results in low-frequency changes in pixel values, by filtering out the low-frequency information, the high pass pre-filter module can highlight the subtle changes, making the tampering behaviour easier to be detected. The high pass filter feeds each frame with one pixel, and the relationship between the frames is modelled by calculating the residuals between the target frame and its neighbouring incoming frames. The feature extraction module is then constructed to learn features from the video residual frames using two ResNet blocks connected in series, which contain two bottleneck units, each consisting of three consecutive convolutional layers and an identity hopping connection, where the kernel sizes of the convolutional layers are 1 × 1, 3 × 3, and 1 × 1, respectively; convolutional spans of 1, and the last convolutional layer with a span of 2, are for spatial pooling; instance normalisation and ReLu activation are performed before each convolution. The proposed network is a fully convolutional network without fully connected layers and thus can handle videos of arbitrary size.

2.2. Flow-guided inter-frame residual stream

The results obtained through the high pass filter can be disturbed by video motion, so the FGIFR module is designed as shown in Fig. 2, where the Frame alignment module (FAM) is the frame alignment module, the main role of this module is to ensure that the frames in a video sequence are consistent in time for subsequent analysis or processing. In video processing, there may be small offsets or variations between different frames due to camera movement, vibration or scene changes. By detecting and calibrating these offsets, the frame alignment module enables the frames in a video sequence to be temporally consistent, thereby improving the accuracy and stability of the subsequent processing steps. The FGIFR module is designed to learn the motion information between frames and use this information to generate more accurate interpolated frames. The flow field is first estimated from the flow network Ƒ (FlowNet) based on the alignment of optical flow pairs and neighbouring video frames:

$$\mathrm{m}_{t\rightarrow t+1} = \int(\mathrm{I}_t, \mathrm{I}_{t+1}) \tag{1}$$

It is assumed that the flow vector of pixel $\mathrm{I}_t(\mathrm{x}, \mathrm{y})$ of It in is $M_{t\rightarrow t+1}(x, y) = (\Delta x, \Delta y)$, where the pixel of $\mathrm{I}(\mathrm{x}, \mathrm{yt}+1'')$ is It and It+1 and the neighbouring video frame I satisfy the conditions:

$$(\mathrm{x}', \mathrm{y}') = ([\mathrm{x} + \Delta\mathrm{x}], [\mathrm{y} + \Delta\mathrm{y}]) \tag{2}$$

The pixels in the middle I_{t+1} are then projected to a new location to form the virtual frame $\hat{I}_{t+1}$, and the following equation is satisfied:

$$\begin{cases}\hat{I}_{t+1}(x,y)=I_{t+1}(x',y') & (1)\\ \hat{I}_{t+1}(x,y)=0 & (2)\end{cases} \tag{3}$$

Where (1) satisfies conditions $0 \le x' < w, 0 \le y' < h$, w, and h are the width and height of the frame, respectively, by which a virtual frame I_{t+1} is generated from a based on $m_{t \to t+1}$ to compute the residuals between frames to satisfy the conditions:

$$R^{fl} = \{I_t - \hat{I}_{t-1}, I_t - \hat{I}_{t+1}\} \tag{4}$$

After that, two ResNet blocks connected in series are used with the same structure as the ResNet blocks connected in series with the Intra-frame residual stream, and then the feature maps from the Intra-frame residual stream and the Flow-guided inter-frame residual stream are connected and fed into the other two ResNet blocks, finally outputting a 1024-channel feature map with a spatial resolution of 1/16 of the input video frame. The feature maps from the Intra-frame residual stream and Flow-guided inter-frame residual stream data streams are then connected and fed into the other two ResNet blocks, finally outputting a 1024-channel feature map with a spatial resolution of 1/16 of the input video frame.

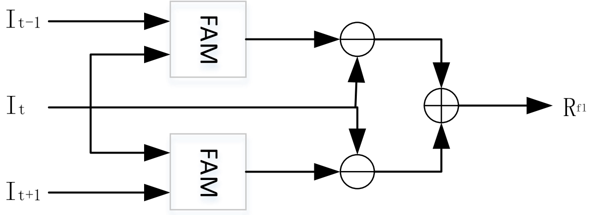

Figure 2: Optical flow guided interframe residual module

2.3. Decoder Networks

Since the video has time domain information, the correlation between consecutive frames is exploited to improve the localisation performance using Bi-ConvolLSTM (Bi-ConvolLSTM) and is embedded in a transposed convolution-based decoder network as shown in Fig. 3. The combination of the two structures allows for the fusion of both temporal and spatial information in the decoder, leading to a better understanding of the input sequences and generation of the corresponding outputs. Each transpose convolution is performed 4 times upscaling and the final output should be the same size as the input frame, before each transpose convolution three consecutively learnt features are used as input using LSTM, one to process the input sequence according to the time forward and the other to process the input sequence according to the time inverse taking into account both past and future information of the input sequence. Finally to reduce artefacts a 5 × 5 convolution was performed to output the prediction mask.

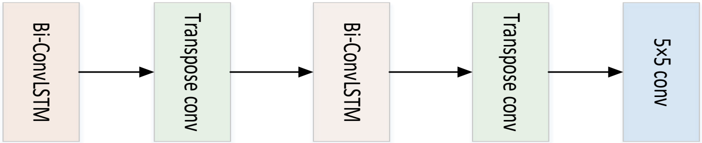

Figure 3: Decoder network

2.4. Refining the network

The purpose of the refinement network is to reduce redundant computation and highlight details, although the base network operates on each frame, the video object is not as effective at the edges of the predictive mask output by the base network when it is in motion, and artefacts appear. The accurate object mask output by the base network is used to guide the VOS model for segmentation, which can help the model locate and segment the object more quickly, reduce unnecessary searches, and present clear object boundaries to improve the model performance. First, the VOS model will perform feature extraction on the video frames to capture the semantic and spatial information of each pixel. Next, the VOS model will use the results of the feature extraction to segment the objects in each pixel of the video frame through a pixel-level classification or segmentation algorithm. Once the objects in the video frame are segmented, the VOS model will track the objects using a tracking algorithm based on the location and shape information of the objects, and the VOS model will consider the spatiotemporal consistency of the objects in the video sequence, i.e., to maintain the temporal and spatial continuity and consistency of the object segmentation results.

2.5. Loss function

Let $L_{\text{base}-refiner}$ denote the total loss function defined as follows:

$$L_{\text{base}-refiner} = \lambda_1 L_{base} + \lambda_2 L_{refiner} \tag{5}$$

Where λ_1 and λ_2 are weighting factors, for the loss in the underlying network is used to deal with the imbalance using the optimised dice loss, where the dice loss is defined as follows:

$$L_{base} = 1 - \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2} \tag{6}$$

where N is the total number of pixels and P_i and g_i are the predicted and ground truth labels, respectively. For the loss $L_{refiner}$ ofin the G_{refiner} refinement network the cross-entropy loss is used.

3. Experimental

3.1. Implementation details

The training and test data were constructed by using different datasets. The training datasets consist of DAVIS2016 and DAVIS2017, where the DAVIS2016 dataset contains 50 video sequences totalling 2,500 frames and the DAVIS2017 dataset contains 60 video sequences totalling 3,455 frames. Each video sequence in both datasets contains consecutive image frames, as well as pixel-level foreground segmentation annotations. The datasets cover a variety of scenes and actions, including people, animals, and natural landscapes. The test data was obtained using VideoMatte240K by selecting 50 4K video data from VideoMatte240K, a dataset enriched with a variety of motion types and moving objects, each with a duration of 15 to 30 seconds, and using the chroma keying tool, Adobe After Effects, to generate different foreground masks and foreground frames, to obtain a more accurate prediction of the masks as reference objects.

In this case, the VOS is using the pre-trained STCN, which leads to an unavoidable increase in complexity, total training time and memory loss due to the fact that the VOS is executed after the G_{base} network. The proposed network is implemented using Tensorflow, and during training, the Adam optimiser is used with the initial learning rate set to 1 × 10-4, for the weighting parameter in the loss function equation (5) is empirically chosen to be $\lambda_1 = \lambda_2 = 0.1$. Reduced by 50% after each epoch, and Xavier is used as the initialisation kernel weights with a zero initialisation bias, and the L2 regularity is used with the weights decayed to 1 × 10-4 for L2 regularisation. The batch size used for both training and test sets is 4. All experiments are performed on an NVIDIA RTX 3090 (24G), and to evaluate the localisation performance, pixel-level F1 scores and concatenated intersections (IoU) are used as performance metrics. The metrics are calculated independently for each video frame and the average of all tested video frames is reported.

3.2. Ablation experiments

In order to verify the detection enhancement effect of the VOS module on the whole model, especially at the edges, it was tested by testing the datasets DAVIS2016, DAVIS2017, and VideoMatte240K, and the experiments are shown in Table 1, and comparing with Table 2, it can be obtained that the VOS module enhances all the four detections and our model.

Table 1: Test results of DAVIS2016, DAVIS2017 and VideoMatte240K datasets on different methods

Method	DAVIS2016		DAVIS2017		VideoMatte240K	
	IoU-pixel	F_1-pixel	IoU-pixel	F_1-pixel	IoU-pixel	F_1-pixel
CFA	0.2231	0.3010	0.2489	0.3163	0.3145	0.4210
LSTMEnDic	0.2502	0.3426	0.2593	0.3604	0.4093	0.5203
RRUNet	0.3498	0.4225	0.3621	0.4381	0.4751	0.5011
Mantra-Net	0.3625	0.5301	0.4424	0.5431	0.4801	0.5136
Proposed	**0.5952**	**0.7105**	**0.6132**	**0.7212**	**0.6381**	**0.7501**

3.3. Comparison with other methods

Table 2: In class testing on the VideoMate240K dataset

Method	DAVIS2016		DAVIS2017		VideoMatte240K	
	IoU	F_1	IoU	F_1	IoU	F_1
CFA-VOS	0.2468	0.3186	0.2601	0.3265	0.3257	0.4328
LSTMEnDic-VOS	0.2653	0.3572	0.2738	0.3729	0.4215	0.5318
RRUNet-VOS	0.3625	0.4374	0.3752	0.4421	0.4863	0.5138
Mantra-Net-VOS	0.3825	0.5491	0.4584	0.5521	0.4921	0.5275
Proposed-VOS	**0.6032**	**0.7242**	**0.6276**	**0.7342**	**0.6438**	**0.7615**

In this experiment, the performance of our proposed method is compared with existing methods on the datasets DAVIS2016, DAVIS2017, and VideoMatte240K, where all three datasets are processed with video keying to replace the background. Existing methods include the generic image tampering localisation LSTMEnDic, two deep feature-based forensic methods RRUNet, Mantra Net, and the traditional forensic method CFA. The above data is used to train and test models for all methods. We convert the predictions obtained through the different methods into binary graphs by using a series of thresholds and computing the F1 scores and IOUs, the results are shown in Table 2, Fig. 4. It is clear that our method outperforms the four existing methods.

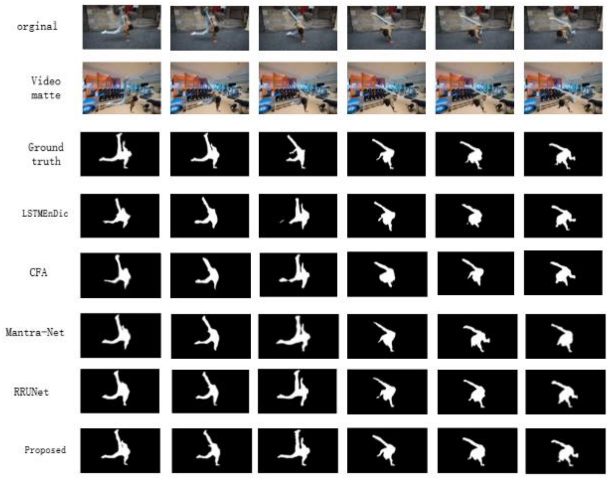

Figure 4: Plot of test results on different methods

3.4. Robust experiments

To further assess the robustness of the method, a series of experiments are conducted on three datasets to test the stability of the model under different attacks using edge pixel-level F1 scores. The experimental results are shown in Table 3 and Fig. 5. From the table and figure it can be seen that these common attack methods have less impact on the experimental results

Table 3: F1 metrics under different attacks

Attack	Parameter	DAVIS2016	DAVIS2017	VideoMatte240K
MPEG Compression	quality=100	0.7126	0.7439	0.7522
	quality=80	0.6920	0.7189	0.7298
	quality=60	0.6873	0.6936	0.7231
Median Blur	kernal size=3	0.6892	0.7192	0.7156
	kernal size=5	0.6639	0.6636	0.6941
	kernal size=7	0.6427	0.6553	0.6830
Resize	Nomanipulation	0.7242	0.7342	0.7615
	factor=0.8	0.6982	0.7082	0.7351
	factor=0.6	0.6794	0.6917	0.7135

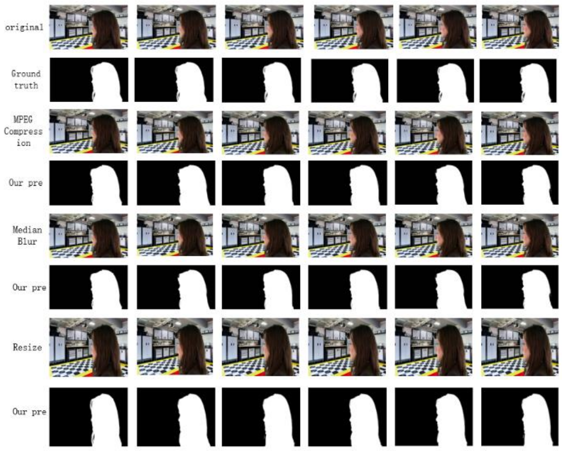

Figure 5: Experimental effect graphs under different attacks

4. Conclusions

In this paper, we propose a method to detect and localise video tampering regions by means of a trace left by time and space domain information. The trace is enhanced by intra-frame and inter-frame residuals, where the inter-frame residuals are really extracted under the guidance of optical flow for better detection of temporal and spatial inconsistencies of tampering. The prediction mask is obtained by exploiting the correlation between sequence frames using bidirectional LSTM and transposed convolution as a decoder network. Afterwards, a VOS network is used to obtain clearer object boundaries. The proposed end-to-end framework can predictively locate the tampered regions by a high pass filter.

References

[1] Stamm, M.C.; Wu, M.; Liu, K.R. Information forensics: An overview of the first decade. IEEE Access 2013,1, 167-200.

[2] J. Redi, W. Taktak, Digital image forensics: a booklet for beginners, MultimediaTools and Applications 51 (2) (2011) 133–162.

[3] B. Mahdian, R. Nedbal, S. Saic, Blind verification of digital image originality: astatistical approach, IEEE Transactions on Information Forensics and Security 8 (9) (2013) 1531–1539.

[4] Asghar, K., Habib, Z., & Hussain, M. (2017). Copy-move and splicing image forgery detection and localization techniques: a review. Australian Journal of Forensic Sciences, 49(3), 281-307.

[5] Sowmya, K., & Chennamma, H. (2015). A survey on video forgery detection. International Journal of Computer Engineering and Applications,9(2), 17-27.

[6] Birajdar, G. K., & Mankar, V. H. (2013). Digital image forgery detection using passive techniques: A survey. Digital investigation, 10(3), 226-245.

[7] Peng Zhou, Ning Yu, Zuxuan Wu, Larry S Davis, Abhinav Shrivastava,and Ser-Nam Lim, "Deep video inpainting detection," arXiv preprintarXiv:2101.11080, 2021.

[8] Xiangling Ding, Yifeng Pan, Kui Luo, Yanming Huang, Junlin Ouyang,and Gaobo Yang, "Localization of deep video inpainting basedon spatiotemporal convolution and refinement network," in IEEEInternational Symposium on Circuits and Systems, 2021, pp. 1–5.

[9] Haodong Li and Jiwu Huang, "Localization of deep inpaintingusing high-pass fully convolutional network," in IEEE InternationalConference on Computer Vision, 2019, pp. 8301–8310.