Chapter 13
Video Frame Deletion and Duplication

Chengjiang Long, Arslan Basharat, and Anthony Hoogs

Videos can be manipulated in a number of different ways, including object addition or removal, deep fake videos, temporal removal or duplication of parts of the video, etc. In this chapter, we provide an overview of the previous work related to video frame deletion and duplication and dive into the details of two deep-learning-based approaches for detecting and localizing frame deletion (Chengjiang et al. 2017) and duplication (Chengjiang et al. 2019) manipulations. This should provide the reader a brief overview of the related research and details of a couple of deep-learning-based forensics methods to defend against temporal video manipulations.

13.1 Introduction

Digital video forgery (Sowmya and Chennamma 2015) is referred to as intentional modification of the digital video for fabrication. A common digital video forgery technique is temporal manipulation, which includes frame sequence manipulations such as dropping, insertion, reordering and looping. By altering only the temporal aspect of the video the manipulation is not detectable by single-image forensic techniques; therefore, there is need for digital forensics methods that perform temporal analysis of videos to detect such manipulations.

In this chapter, we will first focus on the problem of video frame deletion detection in a given, possibly manipulated, video without the original video. As illustrated in Fig. 13.1, we define a frame drop to be a removal of any number of consecutive

C. Long
Meta Reality Labs, Burlingame, CA, USA

A. Basharat (✉) · A. Hoogs
Kitware, Inc., Clifton Park, NY, USA
e-mail: arslan.basharat@kitware.com

H. T. Sencar et al. (eds.), *Multimedia Forensics*, Advances in Computer Vision and Pattern Recognition, https://doi.org/10.1007/978-981-16-7621-5_13

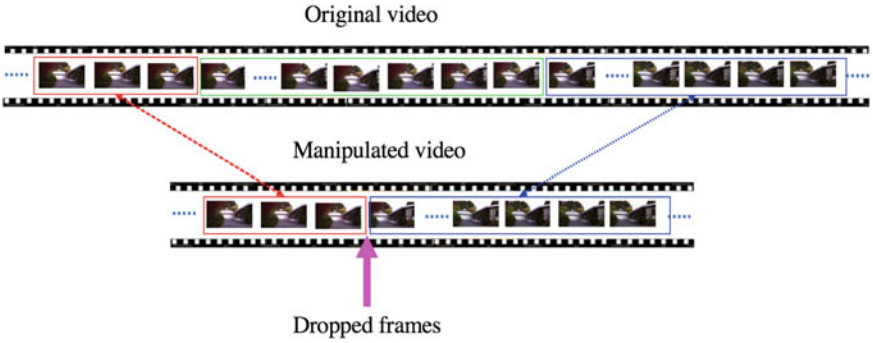

Fig. 13.1 The illustration of frame dropping detection challenge. Assuming that there are three consecutive frame sequences (marked in red, green and blue, respectively) in an original video, the manipulated video is obtained after removing the green frame sequence. Our goal is to identify the location of the frame drop at the end of the red frame sequence and the beginning of the blue frame sequence

frames within a video shot.[1] In our work (Chengjiang et al. 2017) to address this problem, we only consider videos with a single shot to avoid the confusion between frame drops and shot breaks. Single-shot videos are prevalent from various sources, like mobile phones, car dashboard cameras or body-worn cameras.

To the best of our knowledge, only a small amount of recent work (Thakur et al. 2016) has explored automatically detecting dropped frames without a reference video. In digital forgery detection, we cannot assume a reference video, unlike related techniques that detect frame drops for quality assurance. Wolf (2009) proposed a frame-by-frame motion energy cue defined based on the temporal information difference sequence for finding dropped/repeated frames, among which the changes are slight. Unlike Wolf's work, we detect the locations where frames are dropped in a manipulated video without being compared with the original video. Recently, Thakur et al. (2016) proposed an SVM-based method to classify tampered or non-tampered videos. In this work, we explore the authentication (Valentina et al. 2012; Wang and Farid 2007) of the scene or camera to determine if a video has one or more frame drops without a reference or original video. We expect such authentication is able to explore underlying spatio-temporal relationships across the video so that it is robust to digital-level attacks and conveys a consistency indicator across the frame sequences.

We believe that we can still use similar assumption that consecutive frames are consistent with each other and the consistency will be destroyed if there exists temporal manipulation. To authenticate a video, two-frame techniques such as color histogram, motion energy (Stephen 2009) and optical flow (Chao et al. 2012; Wang et al. 2014) have been used. By only using two frames these techniques cannot generalize to work on both videos with rapid scene changes (often from fast camera

[1] A shot is a consecutive sequence of frames captured between the start and stop operations of a single video camera.

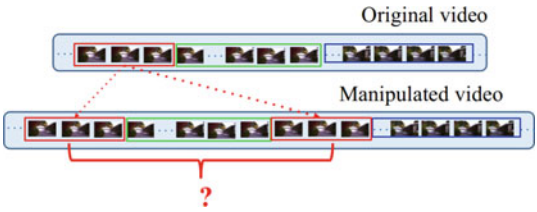

Fig. 13.2 An illustration of frame duplication manipulation in a video. Assume an original video has three sets of frames indicated here by red, green and blue rectangles. A manipulated video can be generated by inserting a second copy of the red set in the middle of the green and the blue sets. Our goal is to detect both instances of the red set as duplicated and also determine that the second instance is the one that's forged

motion) and videos with subtle scene changes such as static camera surveillance videos.

In the past few years, deep learning algorithms have made significant breakthroughs, especially in the image domain (Krizhevsky et al. 2012). The features computed by these algorithms have been used for image matching/classification (Zhang et al. 2014; Zhou et al. 2014). In this chapter, we evaluate approaches using these features for dropped frame detection using two to three frames. However, these image-based deep features still lack modeling the motion effectively.

Inspired by Tran et al.'s C3D network (Tran et al. 2015), which is able to extract powerful spatio-temporal features for action recognition, we propose a C3D-based network for detecting frame drops, as illustrated in Fig. 13.3. As we can observe, there are three aspects to distinguish our C3D-based network approach (Chengjiang et al. 2017) from Tran et al.'s work. (1) Our task is to check whether there exist frames dropped between the 8th and the 9th frame, which makes the center part more informative than the two ends of the 16-frame video clips; (2) the output of the network has two branches, which correspond to "frame drop" and "no frame drop", between the 8th and the 9th frame; (3) unlike most approaches, we use the output scores from the network as confidence score directly and define confidence score with a peak detection step and a scale term based on the output score curves; and (4) such a network is able to not only predict whether the video has frame dropping but also detect the exact location where the frame dropping occurs.

To summarize, the contributions of our work (Chengjiang et al. 2017) are:

- Proposed a 3D convolutional network for frame dropping detection, and the confidence score is defined with a peak detection step and a scale term based on the output score curves. It is able to identify whether there exists frame dropping and even determine the exact location of frame dropping without any information of the reference/original video.
- For performance comparison, we also compared a series of baselines, including cue-based algorithms (color histogram, motion energy and optical flow) and learning-based algorithms (an SVM algorithm and convolutional neural networks (CNNs) using two or three frames as input).

- The experimental results on both the Yahoo Flickr Creative Commons 100 Million (YFCC100m) dataset and the Nimble Challenge 2017 dataset clearly demonstrate the efficacy of the proposed C3D-based network.

An increasingly large volume of digital video content is becoming available in our daily lives through the internet due to the rapid growth of increasingly sophisticated, mobile and low-cost video recorders. These videos are often edited and altered for various purposes using image and video editing tools that have become more readily available. Manipulations or forgeries can be done for nefarious purposes to either hide or duplicate an event or content in the original video. Frame duplication refers to a video manipulation where a copy of a sequence of frames is inserted into the same video either replacing previous frames or as additional frames. Figure 13.2 provides an example of frame duplication where in the manipulated video the red frame sequence from the original video is inserted between the green and the blue frame sequences. As a real-world example, frame duplication forgery could be done to hide an individual leaving a building in a surveillance video. If such a manipulated video was part of a criminal investigation, without effective forensics tools the investigators could be misled.

Videos can also be manipulated by duplicating a sequence of consecutive frames with the goal of concealing or imitating specific content in the same video. In this chapter, we also describe a coarse-to-fine framework based on deep convolutional neural networks to automatically detect and localize such frame duplication (Chengjiang et al. 2019). First, an I3D network finds coarse-level matches between candidate duplicated frame sequences and the corresponding selected original frame sequences. Then a Siamese network based on ResNet architecture identifies fine-level correspondences between an individual duplicated frame and the corresponding selected frame. We also propose a robust statistical approach to compute a video-level score indicating the likelihood of manipulation or forgery. Additionally, for providing manipulation localization information we develop an inconsistency detector based on the I3D network to distinguish the duplicated frames from the selected original frames. Quantified evaluation on two challenging video forgery datasets clearly demonstrates that this approach performs significantly better than four state-of-the-art methods.

It is very important to develop robust video forensic techniques, to catch videos with increasing sophisticated forgeries. Video forensics techniques (Milani et al. 2012; Wang and Farid 2007) aim to extract and exploit features from videos that can distinguish forgeries from original, authentic videos. Like other areas in information security, the sophistication of attacks and forgeries continue to increase for images and videos, requiring a continued improvement in forensic techniques. Robust detection and localization of duplicated parts of a video can be a very useful forensic tool for those tasked with authenticating large volumes of video content.

In recent years, multiple digital video forgery detection approaches have been employed to solve this challenging problem. Wang and Farid (2007) proposed a frame duplication detection algorithm which takes the correlation coefficient as a measure of similarity. However, such an algorithm results in a heavy computational load due to a

large number of correlation calculations. Lin et al. (2012) proposed to use histogram difference (HD) instead of correlation coefficients as the detection features. The drawback is that the HD features do not show strong robustness against common video operations or attacks. Hu et al. (2012) propose to detect duplicated frames using video sub-sequence fingerprints extracted from the DCT coefficients. Yang et al. (2016) propose an effective similarity-analysis-based method that is implemented in two stages, where the features are obtained via SVD. Ulutas et al. propose to use a BoW model (Ulutas et al. 2018) and binary features (Ulutas et al. 2017) for frame duplication detection. Although deep learning solutions, especially those based on convolution neural networks, have demonstrated promising performance in solving many challenging vision problems such as large-scale image recognition (Kaiming et al. 2016; Stock and Cisse 2018), object detection (Shaoqing et al. 2015; Yuhua et al. 2018; Tang et al. 2018) and visual captioning (Venugopalan et al. 2015; Aneja et al. 2018; Huanyu et al. 2018), no deep learning solutions were developed for this specific task at the time, which motivated us to fill this gap.

In Chengjiang et al. (2019) we describe a coarse-to-fine deep learning framework, called C2F-DCNN, for frame duplication detection and localization in forged videos. As illustrated in Fig. 13.4, we first utilize an I3D network (Carreira and Zisserman 2017) to obtain the candidate duplicate sequences at a coarse level; this helps narrow the search faster through longer videos. Next, at a finer level, we apply a Siamese network composed of two ResNet networks (Kaiming et al. 2016) to further confirm duplication at the frame level to obtain accurate corresponding pairs of duplicated and selected original frames. Finally, the duplicated frame range can be distinguished from the corresponding selected original frame range by our inconsistency detector that is designed as an I3D network with 16-frames as an input video clip.

Unlike other methods, we consider the consistency between two consecutive frames from a 16-frame video clip in which these two consecutive frames are at the center, i.e., 8th and 9th frame. This is aimed at capturing the temporal context for matching a range of frames for duplication. Inspired by Long et al. (2017), we design an inconsistency detector based on the I3D network to cover three categories, i.e., "none", "frame drop" and "shot break", which represent that between the 8th and 9th frame there are no manipulations, frames removal within one shot, and a shot boundary transition, respectively. Therefore, we are able to use output scores from the learned I3D network to formulate a confidence score of inconsistency between any two consecutive frames to distinguish the duplicated frame range from the selected original frame range, even in videos with multiple shots.

We also proposed a heuristic strategy to produce a video-level frame duplication likelihood score. This is built upon the measures like the number of possible frames duplicated, the minimum distance between duplicated frames and selected frames, and the temporal gap between the duplicated frames and the selected original frames.

To summarize, the contributions of this approach (Chengjiang et al. 2019) are as follows:

- A novel coarse-to-fine deep learning framework for frame duplication detection and localization in forged videos. This framework features fine-tuned I3D networks

and the ResNet Siamese network, providing a robust yet efficient approach to process large volumes of video data.
- Designed an inconsistency detector based on a fine-tuned I3D network that covers three categories to distinguish duplicated frame range from the selected original frame range.
- A heuristic formulation for video-level detection score, which leads to significant improvement in detection benchmark performance.
- Evaluated performance on two video forgery datasets and the experimental results strongly demonstrate the effectiveness of the proposed method.

13.2 Related Work

13.2.1 Frame Deletion Detection

The most related prior work can be roughly split into two categories: *video inter-frame forgery identification* and *shot boundary detection*.

Video inter-frame forgery identification. Video inter-frame forgery involves frame insertion and frame deletion. Wang et al. proposed an SVM method (Wang et al. 2014) based on the assumption that the optical flows are consistent in an original video, while in forgeries the consistency will be destroyed. Chao's optical flow method (Chao et al. 2012) provides different detection schemes for inter-frame forgery based on the observation that the subtle difference between frame insertion and deletion. Besides optic flow, Wang et al. (2014) also extracted the consistency of correlation coefficients of gray values as distinguishing features to classify original videos and forgeries. Zheng et al. (2014) proposed a novel feature called block-wise brightness variance descriptor (BBVD) for fast detection of video inter-frame forgery. Different from this inter-frame forgery identification, our proposed C3D-based network (Chengjiang et al. 2017) is able to explore the powerful spatio-temporal relationships as the authentication of the scene or camera in a video for frame dropping detection.

Shot Boundary Detection. There is a large amount of work to solve the shot boundary detection problem (Smeaton et al. 2010). The task of shot boundary detection (Smeaton et al. 2010) is to detect the boundaries to separate multiple shots within a video. The TREC video retrieval evaluation (TRECVID) is an important benchmark dataset for automatic shot boundary detection challenge. And different research groups from across the world have worked to determine the best approaches to shot boundary detection using a common dataset and common scoring metrics. Instead of detecting where two shots are concatenated, we are focused on detecting a frame drop within a single shot.

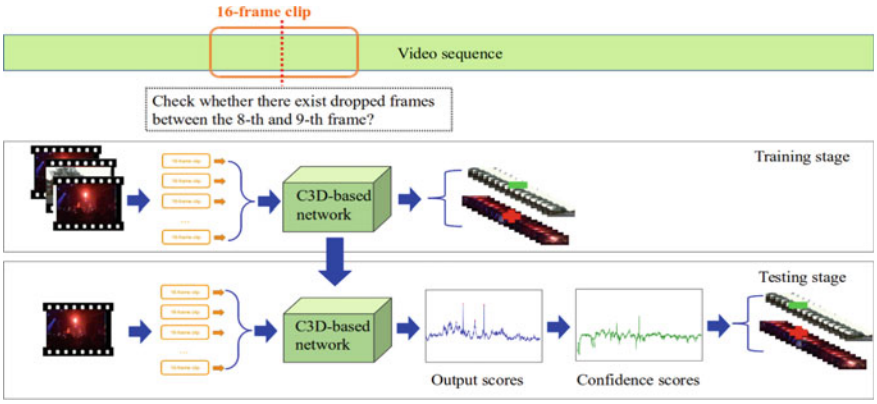

Fig. 13.3 The pipeline of the C3D-based method. At the training stage, the C3D-based network takes 16-frame video clips extracted from the video dataset as input, and produces two outputs, i.e., "frame drop" (indicated with "+") or "no frame drop" (indicated with "–"). At the testing stage, we decompose a testing video into a sequence of continuous 16-frame clips and then fit them into the learned C3D-based network to obtain the output scores. Based on the score curves, we use a peak detection step and introduce a scale term to define the confidence scores to detect/identify whether there exist dropped frames for per frame clip or per video. The network model consisted of 66 million parameters with $3 \times 3 \times 3$ filter size at all convolutional layers

13.2.2 Frame Duplication Detection

The research related to frame duplication can be broadly divided into *inter-frame forgery*, *copy-move forgery* and *convolutional neural networks*.

Inter-frame forgery refers to frame deletion and frame duplication. For features used for inter-frame forgery, either spatially or temporally, keypoints are extracted from nearby patches recognized over distinctive scales. Keypoint-based methodologies can be further subdivided into direction-based (Douze et al. 2008; Le et al. 2010), keyframe-based coordinating (Law-To et al. 2006) and visual-words-based (Sowmya and Chennamma 2015). In particular, keyframe-based feature has been shown to perform well for close video picture/feature identification (Law-To et al. 2006).

In addition to keypoint-based features, Wu et al. (2014) propose a velocity field consistency-based approach to detect inter-frame forgery. This method is able to distinguish the forgery types, identify the tampered video and locate the manipulated positions in forged videos as well. Wang et al. (2014) propose to make full use of the consistency of the correlation coefficients of gray values to classify original videos and inter-frame forgeries. They also propose an optical flow method (Wang et al. 2014) based on the assumption that the optical flows are consistent in an original video, while in forgeries the consistency will be destroyed. The optical flow is extracted as a distinguishing feature to identify inter-frame forgeries through a support vector machine (SVM) classifier to recognize frame insertion and frame deletion forgeries.

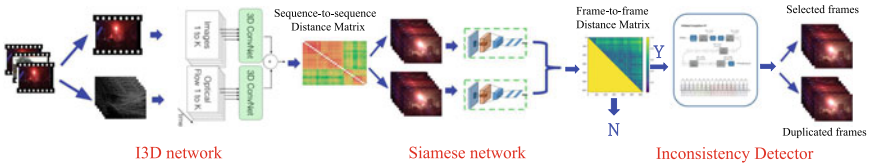

Fig. 13.4 The C2F-DCNN framework for frame duplication detection and localization. Given a testing video, we first run the I3D network (Carreira and Zisserman 2017) to extract deep spatial-temporal features and build the coarse sequence-to-sequence distance to determine the possible frame sequences that are likely to have frame duplication. For the likely duplicated sequences, a ResNet-based Siamese network further confirms a frame duplication at the frame level. For the videos with duplication detected, temporal localization is determined with an I3D-based inconsistency detector to distinguish the duplicated frames from the selected frames

Huang et al. (2018) proposed a fusion of audio forensics detection methods for video inter-frame forgery. Zhao et al. (2018) developed a similarity analysis-based method to detect inter-frame forgery in a video shot. In this method, the HSV color histogram is calculated to detect and locate tampered frames in the shot, and then the SURF feature extraction and FLANN (Fast Library for Approximate Nearest Neighbors) matching are used for further confirmation.

Copy-move forgery is created by copying and pasting content within the same frame, and potentially post-processing it (Christlein et al. 2012; D'Amiano et al. 2019). Wang et al. (2009) propose a dimensionality reduction approach through principal component analysis (PCA) on the different pieces. Mohamadian et al. (2013) develop a singular value decomposition (SVD) based method in which the image is isolated into numerous little covering squares and after that SVD is requested to remove the copied frames. Yang et al. (2018) proposed a copy-move forgery detection based on a modified SIFT-based detector. Wang et al. (2018) presented a novel block-based robust copy-move forgery detection approach using invariant quaternion exponent moments. D'Amiano et al. (2019) proposed a dense-field method with a video-oriented version of PatchMatch for the detection and localization of copy-move video forgeries.

Convolutional neural networks (CNNs) have been demonstrated to learn rich, robust and powerful features for large-scale video classification (Karpathy et al. 2014). Various 3D CNN architectures (Tran et al. 2015; Carreira and Zisserman 2017; Hara et al. 2018; Xie et al. 2018) have been proposed to explore spatio-temporal contextual relations between consecutive frames for representation learning. Unlike the existing methods for inter-frame forgery and copy-move forgery which mainly use hand-crafted features or bag-of-words, we take advantage of convolutional neural networks to extract spatial and temporal features for frame duplication detection and localization.

13.3 Frame Deletion Detection

There is limited work exploring frame deletion or dropping detection problem without reference or original video. Therefore, we first introduce a series of baselines, including cue-based and learning-based methods, and then introduce our proposed C3D-based CNN.

13.3.1 Baseline Approaches

We studied three different cue-based baseline algorithms from the literature, i.e., (1) color histogram, (2) optical flow (Wang et al. 2014; Chao et al. 2012) and (3) motion energy (Stephen 2009) as follows:

- **Color histogram**. We calculate the histograms on all R, G and B three channels. Whether there are frames dropped between the two consecutive frames is detected by thresholding the score calculated by the L2 distances based on the color histograms of the two adjacent frames.
- **Optical flow**. We calculate the optical flow (Wang et al. 2014; Chao et al. 2012) from the two adjacent frames by the Lucas-Kanade method. Whether there exist frames dropped between the current frame and the next frame is detected by thresholding the L2 distance between the average moving direction between the previous frame and the current frame, and the average moving direction between the current frame and the next frame.
- **Motion energy**. Motion energy is the temporal information (TI) difference sequence (Stephen 2009), i.e., the difference of Y channel in the YCrCb color space. Whether there exist frames dropped between the current frame and the next frame is detected by thresholding the motion energy between the current frame and the next frame.

Note that each algorithm mentioned above compares two consecutive frames and estimates whether there are missing frames between them. We also developed four learning-based baseline algorithms as follows:

- **SVM**. We train an SVM model to predict whether there are frames dropped between two adjacent frames. The feature vector is the concatenation of the absolute difference of color histograms and the two-dimensional absolute difference of the optical flow directions. The optical flow dimensionality is much smaller than the color histogram, and therefore we give it a higher weight.
- **Pairwise Siamese Network**. We train a Siamese CNN that determines if the two input frames are consecutive or if there is frame dropping between them. Each CNN consists of two convolutional layers and three fully connected layers. The loss used is contrastive loss.
- **Triplet Siamese Network**. We extend the pairwise Siamese network to use three consecutive frames. Unlike the pairwise Siamese network, the triplet Siamese

Table 13.1 A list of related algorithms for temporal video manipulation detection. The first three algorithms are cue-based without any training work. The rest are learned-based algorithms, including the traditional SVM, the popular CNNs and the method we proposed in Chengjiang et al. (2019)

Method	Brief description	Learning?
Color histogram	RGB 3 channel histograms + L2 distance	No
Optical flow	The optic flow (Wang et al. 2014; Chao et al. 2012) with Lucas-Kanade method + L2 distance	No
Motion energy	Based on temporal information difference (Stephen 2009) sequence	No
SVM	770-D feature vector (3×256-D RGB histogram + 2-D optic flow)	Yes
Pairwise Siamese Network	Siamese network architecture (2 conv layers + 3 fc layers + contrastive loss)	Yes
Triplet Siamese Network	Siamese network architecture (Alexnet-variant + Euclidean&contrastive loss)	Yes
Alexnet (Krizhevsky et al. 2012) Network	Alexnet-variant network architecture	Yes
C3D-based Network (Chengjiang et al. 2019)	C3D-variant network architecture + confidence score	Yes

network consisted of three Alexnets (Krizhevsky et al. 2012) merging their output with Euclidean loss between the previous frame and the current frame, and with contrastive loss between the current frame and the next frame.

- **Alexnet-variant Network**. The input frames are converted to gray-scale and put into the RGB channels.

To facilitate the comparison of the competing algorithms, we summarize the above descriptions in Table 13.1.

13.3.2 C3D Network for Frame Deletion Detection

The baseline CNN algorithms we investigated lacked a strong temporal feature suitable to capture the signature of frame drops. These algorithms only used features from two to three frames that were computed independently. C3D network was originally designed for action recognition, however, we found that spatio-temporal signature

produced by the 3D convolution is also very effective in capturing the frame drop signatures.

The pipeline of our proposed method is as shown in Fig. 13.3. As we can observe, there are three modifications from the original C3D network. First, the C3D network takes clips of 16 frames, therefore we check the center of the clip (between frames 8 and 9) for frame drops to give equal context on both sides of the drop. This is done by formulating our training data so that frame drops only occur in the center. Secondly, we have a binary output associated with "frames dropped" and "no frames dropped" between the 8th and 9th frame. Lastly, we further refine the per-frame network output scores into a confidence score using peak detection and temporal scaling to further suppress the noisy detections. With the refined confidence scores we are not only able to identify whether the video has frame drops but also localize them by applying the network to the video in a sliding window fashion.

13.3.2.1 Data Preparation

To obtain the training data, we used 2,394 iPhone 4 consumer videos from the World Dataset made available on the DARPA Media Forensics (MediFor) program for research. We pruned the videos such that all videos were of length 1–3 min. We get ended up with 314 videos, of which we randomly selected 264 videos for training, and the rest 50 videos for validation. We developed a tool that randomly drops fixed-length frame sequences from videos. It picks a random number of frame drops and random frame offsets in the video for each removal. The frame drops do not overlap, and it forces 20 frames to be kept around each drop. In our experiments, we manipulate each video many different times to create more data. We vary the fixed frame drop length to see how it affects detection we used 0.5 s, 1 s, 2 s, 5 s and 10 s as five different frame drop durations. We used the videos with these drop durations to train a general C3D-based network for frame drop detection.

13.3.2.2 Training

We use momentum $\mu = 0.9, \gamma = 0.0001$ and set power to be 0.075. We start training at a base learning rate of $\alpha = 0.001$ and the "inv" as the learning rate policy. We set the batch size to be 15 and use the 206000th iteration as the learned model for testing, which achieves about 98.2% validation accuracy.

13.3.2.3 Testing

The proposed C3D-based network is able to identify the temporal removal manipulation due to dropped frames in a video and also localize one or more frame drops within the video. We observe that some videos captured by moving digital cameras may have multiple changes due to quickly camera motion, zooming in/out, etc.,

which can be deceiving to the C3D-based network and can result in false frame dropping detections. In order to reduce such false alarms and increase the generalization ability of our proposed network, we propose an approach to refine the raw network output scores to the confidence scores using peak detection and introduction of a scale term based on the output score variation, i.e.,

1. We first detect the peaks on the output score curve obtained from the proposed C3D-based network per video. Among all the peaks, we only pick the top 2% peaks and ignore the rest of the peaks. Then we shift the time window to check the number of peaks (denoted as n_p) appearing in the time window with ith frame as the center (denoted as $W(i)$). If the number is more than one, i.e., other peaks in the neighborhood, the output score $f(i)$ will be penalized. The value will be penalized more if there are a lot of high peaks detected. The intuition behind is that we want to reduce the false alarms when there are multiple peaks occurring close just because the camera is moving or even zooming in/out.
2. We also introduce a scale term $\Delta(i)$ defined as the difference of the median score and the minimum score within the time window $W(i)$ to control the influence of the camera motion.

Based on the above statement, we can obtain the confidence score for the ith frame as

$$f_{conf}(i) = \begin{cases} f(i) - \lambda\Delta(i) & \text{when } n_p < 2 \\ \frac{f(i)}{n_p} - \lambda\Delta(i) & \text{otherwise} \end{cases}, \tag{13.1}$$

where

$$W(i) = \{i - \frac{w}{2}, \ldots, i + \frac{w}{2}\}. \tag{13.2}$$

Note that λ in Eq. 13.1 is a parameter to control how much the scale term affects the confidence score, and w in Eq. 13.2 indicates the width of the time window.

For testing per frame, say ith frame, we first form a 16-frame video clip and set the ith frame to be the 8th frame in the video clip, and then we can get the output score $f_{conf}(i)$. If $f_{conf}(i) > Threshold$, then we predict there are dropped frames between the ith frame and the $(i+1)$th frame. For testing per video, we take it as a binary classification and confidence measure per video. To simplify, we use a simple confidence measure, i.e., $\max_i f_{conf}(i)$ across all frames. If $\max_i f_{conf}(i) > Threshold$, then there are temporal removal within the video. Otherwise, the video is predicted without any temporal removal. The results reported in this work are without any $Threshold$ as we are reporting the ROC curves.

13.3.3 Experimental Result

We conducted the experiments on a Linux machine with Intel(R) Xeon(R) CPU E5-2687 0 @ 3.10 GHz, 32 GB system memory and graphical card NVIDIA GTX 1080

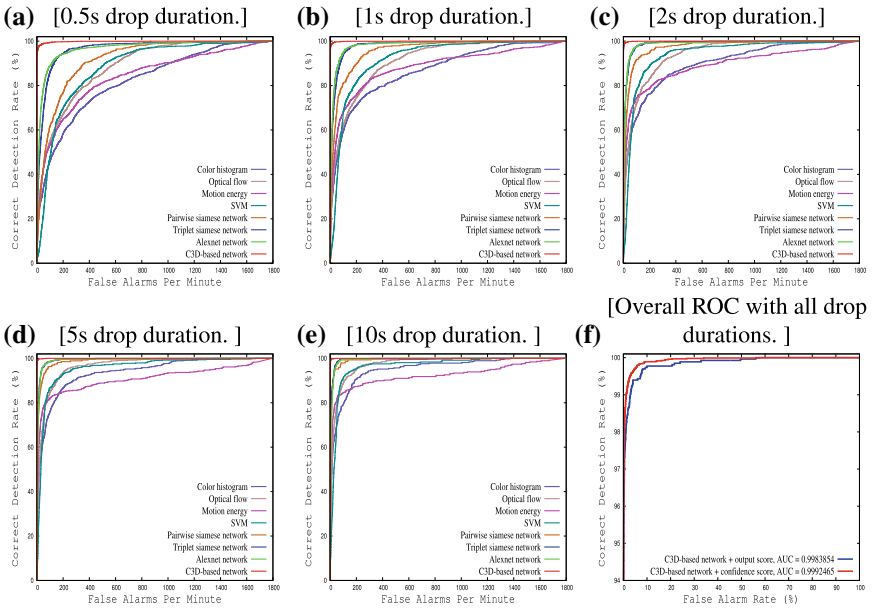

Fig. 13.5 Performance comparison on the YFCC100m dataset against seven baseline approaches, using per-frame ROCs for five different drop durations (**a**–**e**), and (**f**) is frame-level ROC for all the five drop durations combined

(Pascal). We report our results as the ROC curves based on the output score $f_{conf}(i)$ and accuracy as metrics. We present the ROC curves with false positive rate as well as false alarm rate per minute to provide and demonstrate the level of usefulness for a user that might have to adjudicate each detection reported by the algorithm. We present the ROC curves for both per-frame analysis where the ground truth data is available and per-video analysis otherwise.

To demonstrate the effectiveness of the proposed approach, we ran experiments on the YFCC100m dataset[2] and the Nimble Challenge 2017 (Development 2 Beta 1) dataset.[3]

13.3.3.1 YFCC100m Dataset

We download 53 videos tagged with iPhone from Yahoo Flickr Creative Commons 100 Million (YFCC100m) dataset and manually verified that they are single-shot videos. To create ground truth we used our automatic randomized frame dropping tool to generate the manipulated videos. For each video we generated manipulated

[2] YFCC100m dataset: http://www.yfcc100m.org.

[3] Nimble Challenge 2017 dataset: https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation.

Table 13.2 The detailed results of our proposed C3D-based network. $\#_{pos}$ and $\#_{neg}$ are the number instances for the positive and the negative testing 16-frame video clips, respectively. The Acc_{pos} and Acc_{neg} are the corresponding accuracy. Acc is the total accuracy. All the accuracies use the unit %

Duration (s)	$\#_{pos} : \#_{neg}$	Acc_{pos}	Acc_{neg}	Acc
0.5	2816:416633	98.40	98.15	98.16
1	2333:390019	99.49	98.18	98.18
2	2816:416633	99.57	98.11	98.12
5	1225:282355	99.70	98.17	98.18
10	770:239210	100.00	98.12	98.13

videos with frame drops of 0.5, 1, 2, 5 or 10 s intervals at random locations. For each video and each drop duration, we randomly generate 10 manipulated videos. In this way we collect $53 \times 5 \times 10 = 2650$ manipulated videos as testing dataset.

For each drop duration, we run all the competing algorithms in Table 13.1 on the 530 videos with the parameter setting $w = 16$, $\lambda = 0.22$. The experimental results are summarized in the ROC curves for all these five different drop durations in Fig. 13.5.

One can note that (1) the traditional SVM outperforms the three simple cue-based algorithms; (2) the four convolution neural networks algorithms perform much better than the traditional SVM and all the cue-based algorithms; (3) among all the CNN-based networks, both the triplet Siamese network and the Alexnet-variant network perform similar and better than the pairwise Siamese network; and (4) our proposed C3D-based network performs the best. This provides some empirical support to the hypothesis that the proposed C3D-based method is able to take advantage of the temporal and spatial correlations, while the other CNN-based networks only explore the spatial information in the individual frames.

To better understand the C3D-based network, we provide more experimental details in Table 13.2. With the drop duration increase, both the number of positive and negative testing instances decrease and the positive accuracy keeps increasing. As one might expect, the shorter the frame drop duration, the more difficult it is to detect.

We also merge the results of the C3D-based network with five different drop durations in Fig. 13.5 together to plot a unified ROC curve. For comparison, we also plot another ROC curve that uses the output scores to detect whether there exist frame drops within a testing video. As we can see in Fig. 13.5(f), using output score from the C3D-based network, we can still achieve very good performance to 0.9983854 AUC. This observation can be explained by the fact that the raw phone videos from the YFCC100m dataset have less quick motion, no zooming in/out occurring and even no video manipulations. Also, the manipulated videos are generated in the same way as the generation of training manipulated videos with the same five drop durations. Since there are no overlaps on the video contents between training videos and testing videos, such a good performance demonstrates the power and the generalization

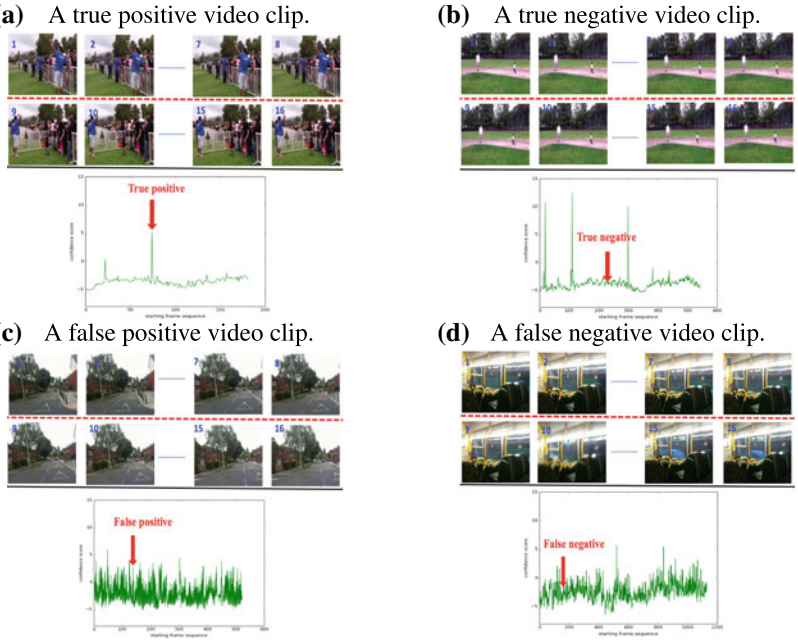

Fig. 13.6 The visualization of two successful examples (one true positive and the other one is true negative) and two failure examples (one false positive and the other one is false negative) from the YFCC100m dataset. The red dashed line indicates the location between the 8th frame and the 9th frame where we test for a frame drop. The red arrows point to the frame on the confidence score plots

ability of the trained network. Although using output score directly achieves a very good AUC, using the confidence score defined in Eq. 13.1 can still improve the AUC from 0.9983854 to 0.9992465. This demonstrates the effectiveness of our confidence score defined with such a peak detection step and a scale term.

We visualize both success and failure cases in our proposed C3D-based network, as shown in Fig. 13.6. Looking at the successful cases in Fig. 13.6(a), "frame drops" is identified correctly in the 16-frame video clip because a man stands at one side in the 8th frame and move to another side suddenly in the 9th frame, and the video clip in Fig. 13.6(b) is predicted as "no frame drops" correctly since a child follows his father in all 16 frames and the 8th frame and the 9th frame are consistent with each other.

Regarding the failures cases, as shown in Fig. 13.6(c), there is no frame drop but it is still identified as "frame drop" between the 8th frame and the 9th frame due to the camera shakes during the video capture of such a street scene. Also, "frame drop" in the top clip cannot be detected correctly between the 8th frame and the 9th frame in the video clip, as shown in Fig. 13.6(d), since the scene inside the bus has almost no visible changes between these two frames.

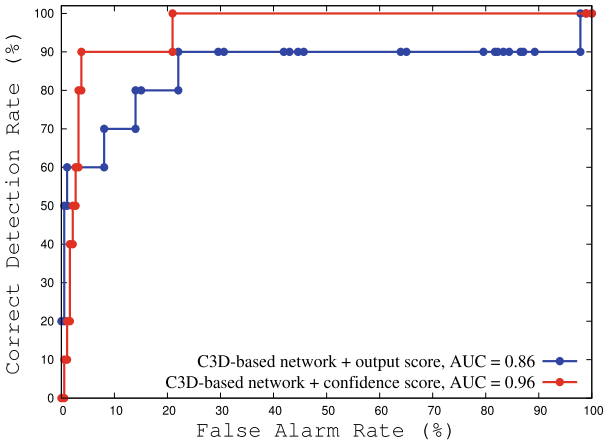

Fig. 13.7 The ROC curve of our proposed C3D-based network on the Nimble Challenge 2017 dataset

Note that our training stage is carried out off-line. Here we only offer the runtime for the testing stage under our experimental environment. For each testing video clip with a 16-frame length, it takes about 2 s. For a one-minute short video with 30 FPS, it requires about 50 min to complete the testing throughout all the frame sequence.

13.3.3.2 Nimble Challenge 2017 Dataset

In order to check whether our proposed C3D-based network is able to identify a testing video with unknown arbitrary drop duration, we also conducted experiments on the Nimble Challenge 2017 dataset, specifically the NC2017-Dev2Beta1 version, in which there are 209 probe videos with various video manipulations. Among these videos, there are six videos manipulated with "TemporalRemove", which is regarded as "frame dropping". Therefore, we run our proposed C3D-based network as a binary classifier to classify all these 209 videos into two groups, i.e., "frame dropping" and "no frame dropping", at the video level. In this experiment, the parameters are set as $w = 500$, $\lambda = 1.25$.

We first plot the output scores from the C3D-based network and the confidence score of each of the six videos is labeled with "TemporalRemove" in Fig. 13.9. It is clear that the video named "d3c6bf5f224070f1df74a63c232e360b.mp4" has the lowest confidence score smaller than zero.

To explain such a case, we further check the content of the video, as shown in Fig. 13.8. As we can observe, this video is even really hard for us to identify it as "TemporalRemoval" since it is taken by a static camera and only the lady's mouth and head are taking very slight changes across the whole video from the beginning to the end. As we trained purely on iPhone videos, our training network was biased

Fig. 13.8 The entire frame sequence of the 34-second video "d3c6bf5f224070f1df74a63c232e360b.mp4", which has 1047 frames and was captured by a static camera. We observe that only the lady's mouth and head are taking very slight change across the video from the beginning to the end

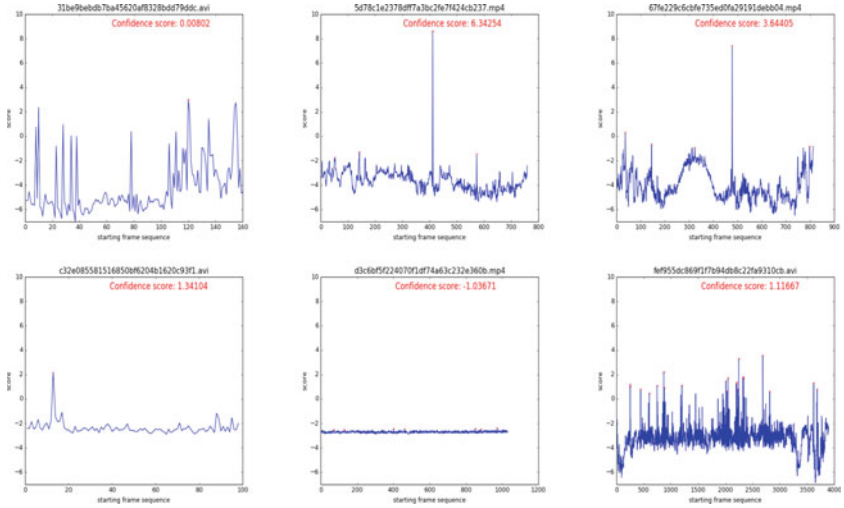

Fig. 13.9 The illustration of output scores from the C3D-based network and their confidence scores for six videos labeled with "TemporalRemove" from the Nimble Challenge 2017 dataset. The blue curve is the output score, the red "+" marks the detected peaks, and the red confidence score is used to determine whether the video can be predicted as a video with "frame drops"

toward videos with camera motion. With a larger dataset of static camera videos, we can train different networks for static and dynamic cameras to address this problem.

We plot the ROC curve in Fig. 13.7. As we can see, the AUC of the C3D-based network with confidence scores is high to 0.96, while the AUC of the C3D-based network with the output scores directly is only 0.86. The insight behind such a significant improvement is that there are testing videos with camera quick-moving, zooming in and out, as well as other types of video manipulations, and our confidence scores defined with the peak detection step and the scale term to penalize multiple peaks occurring too close and large scales is able to significantly reduces the false alarms. Such a significant improvement by 0.11 AUC strongly demonstrates the effectiveness of our proposed method.

13.4 Frame Duplication Detection

As shown in Fig. 13.4, given a probe video, our proposed C2F-DCNN framework is designed to detect and localize frame duplication manipulation. An I3D network is used to produce a sequence-to-sequence matrix and determine the candidate frame sequences at the coarse-search stage. A Siamese network is then applied for a fine-level search to verify whether frame duplications exist. After this, an inconsistency detector is applied to further distinguish duplicated frames from selected frames. All of these steps are described below in detail.

13.4.1 Coarse-Level Search for Duplicated Frame Sequences

In order to efficiently narrow the search space, we start by finding possible duplicate sets of frames throughout the video using a robust CNN representation. We split a video into overlapping frame sequences, where each sequence has 64 frames and the number of overlapped frames is 16. We choose I3D Network (Carreira and Zisserman 2017), instead of using C3D network (Tran et al. 2015) due to these reasons: (1) It inflates 2D ConvNets into 3D and makes filters from typically $N \times N$ square to N×N×N cubic; (2) it bootstraps 3D filters from 2D filters to bootstrap parameters from the pre-trained ImageNet models; and (3) it paces receptive field growth in space, time and network depth.

In this work, we apply the pre-trained off-the-shell I3D network to extract the 1024-dimensional feature vector for $k = 64$ frame sequences since the input for the standard I3D network is 64 rgb-data and 64 flow-data. We observed that a lot of time was being spent on the pre-processing. To reduce the testing runtime, we only compute the first k rgb-data and k flow-data items. For the subsequent frame sequence, we can copy $(k - 1)$ rgb-data and $(k - 1)$ flow-data from the previous video clip, and only calculate the last rgb-data and flow-data. This significantly improved the testing efficiency.

Based on the sequence features, we calculate the sequence-to-sequence distance matrix over the whole video using L2 distance. If the distance is smaller than the threshold T_1, then this indicates that these two frame sequences are likely duplicated and we take them as two candidate frame sequences for further confirmation during the next fine-level search.

13.4.2 Fine-Level Search for Duplicated Frames

For the candidate frame sequences, detected by the previous stage described in Sect. 13.4.1, we evaluate the distance between all pairs of frames across the two sequences, i.e., a duplicated frame and the corresponding selected original frame.

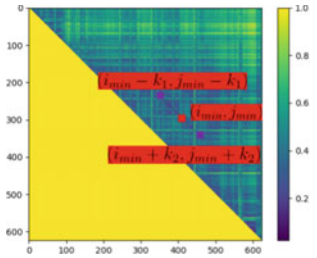

Fig. 13.10 A sample distance matrix based on the frame-to-frame distances computed by the Siamese network between a pair of frame sequences. The symbols shown on the line segment with low distance are used to compute the video-level confidence score for frame duplication detection

For this purpose we propose a Siamese neural network architecture, which learns to differentiate between two frames in the provided pair. It consists of two identical networks by sharing exactly the same parameters, each taking one of the two input frames. A contrastive loss function is applied to the last layers to calculate the distance between the pair. In principle, we can choose any neural network to extract features for each frame.

In this work, we choose the ResNet network (Kaiming et al. 2016) with 152 layers given its demonstrated robustness. We connect two ResNets in the Siamese architecture with a contrastive loss function, and each loss value associated with the distance between a pair of frames is formulated into the frame-to-frame distance matrix, in which the distance is normalized to the range [0, 1]. A distance smaller than the threshold T_2 indicates that these two frames are likely duplicated. For videos that have multiple consecutive frames duplicated we expect to see a line with low values parallel to the diagonal in the visualization of the distance matrix, as plotted in Fig. 13.10.

It is worth mentioning that we provide both frame-level and video-level scores to evaluate the likelihood of frame duplication. For the frame-level score, we can use the value in the frame-to-frame distance directly. For the video-level score, we propose a heuristic strategy to formulate the confidence value. We first find the minimal value of distance $d_{\text{min}} = d(i_{\text{min}}, j_{\text{min}})$ where $i_{\text{min}}, j_{\text{min}} = \text{argmin}_{0 \le i < j \le n} d(i, j)$ is the frame-to-frame distance matrix. Then a search is performed in two directions to find the number of consecutive duplicated frames:

$$k_1 = \underset{k:k \le i_{\text{min}}}{\text{argmax}} \, |d(i_{\text{min}} - k, j_{\text{min}} - k) - d_{\text{min}}| \le \epsilon \tag{13.3}$$

and

$$k_2 = \underset{k:k \le n - j_{\text{min}}}{\text{argmax}} \, |d(i_{\text{min}} + k, j_{\text{min}} + k) - d_{\text{min}}| \le \epsilon \tag{13.4}$$

where $\epsilon = 0.01$ and the length of the interval with duplicated frames can be defined as

$$l = k_1 + k_2 + 1. \tag{13.5}$$

Finally, we can formulate the video-level confidence score as follows:

$$F_{video} = -\frac{d_{\min}}{l \times (j_{\min} - i_{\min})}. \tag{13.6}$$

The intuition here is that a more likely frame duplication is indicated by a smaller value of $d_{\min}$, a longer interval of duplicated frames and a larger temporal gap between the selected original frames and the duplicated frames.

13.4.3 Inconsistency Detector for Duplication Localization

We observe that the duplicated frames inserted into the source video usually yield artifacts due to temporal inconsistency at both the beginning frames and the end frames in a manipulated video. To automatically distinguish the duplicated frames from selected frames, we make use of both spatial and temporal information by training an inconsistency detector to locate this temporal discrepancy. For this purpose, we build upon our work discussed above, Long et al. (2017), which proposed a C3D-based network for frame-drop detection and only works for single-shot videos. Instead of using only one RGB stream data as input, we replace the C3D network with an I3D network to also incorporate the optical flow data stream. It is also worth mentioning that unlike the I3D network used in Sect. 13.4.1, input to the I3D network here is a 16-frame temporal interval, every frame in a sliding window, with RGB and optical flow data. The temporal classification provides insight into the temporal consistency between the 8th and the 9th frame within the 16-frame interval. In order to handle multiple shots in a video with hard cuts, we extend the binary classifier to three classes: "none"—no temporal inconsistency indicating manipulation; "frame drop"—there are frames removed within one-shot video; and "shot break" or "break"—there is a temporal boundary or transition between two video shots. Note that the training data with shot-break videos are obtained from TRECVID 2007 dataset (Kawai et al. 2007), and we only use the hard-cut shot-breaks since soft-cut changes gradually and has strong consistency between any two consecutive frames. The confusion matrix in Fig. 13.11 illustrates the high effectiveness of the proposed I3D network-based inconsistency detector.

Based on the output scores for the three categories from the I3D network, i.e., $S_{I3D}^{none}(i)$, $S_{I3D}^{drop}(i)$ and $S_{I3D}^{break}(i)$, we formulate the confidence score of inconsistency as the following function:

$$S(i) = S_{I3D}^{drop}(i) + S_{I3D}^{break}(i) - \lambda S_{I3D}^{none}(i), \tag{13.7}$$

where λ is the weight parameter, and for the results presented here, we use $\lambda = 0.1$. We assume the selected original frames have a higher temporal consistency with

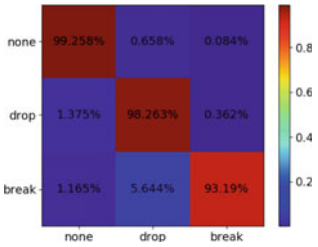

Fig. 13.11 The confusion matrix for three classes of temporal inconsistency within a video, used with the I3D-based inconsistency. We expect a high likelihood of "drop" class at the two ends of the duplicated frame sequence and a high "none" likelihood at the ends of the selected original frame sequence

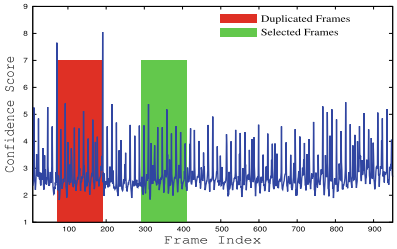

Fig. 13.12 Illustration of distinguishing duplicated frames from the selected frames. The index ranges for the red frame sequence and the green sequence are [72, 191] and [290, 409], respectively. s_1 and s_2 are the corresponding inconsistency scores for the red sequence and green sequence, respectively. Obviously, $s_1 > s_2$, which indicates that the red sequence is duplicated frames as expected

frames before and after such frames than the duplicated frames because the insertion of duplicated frames usually causes a sharp inconsistency at the beginning and the end of the duplicated interval, as illustrated in Fig. 13.12. Given a pair of frame sequences that are potentially duplicated, $[i, i + l]$ and $[j, j + l]$, we compare two scores,

$$s_1 = \sum_{k=-wind}^{wind} S(i - 1 + k) + S(i + l + k) \tag{13.8}$$

and

$$s_2 = \sum_{k=-wind}^{wind} S(j - 1 + k) + S(j + l + k), \tag{13.9}$$

where $wind$ is the window size. We check the inconsistency at both the beginning and the end of the sequence. In this work, we set $wind = 3$ to avoid the failure cases where a few start or end frames were detected incorrectly. If $s_1 > s_2$, then the duplicated frame segment is $[i, i + l]$. Otherwise, the duplicated frame segmentation

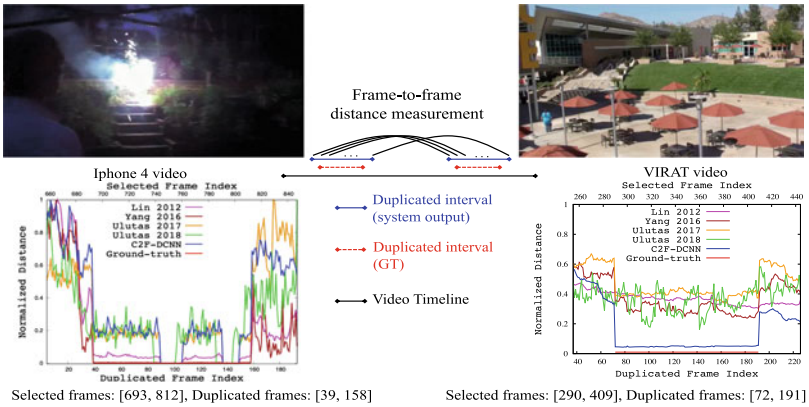

Fig. 13.13 Illustration of frame-to-frame distance between duplicated frames and the selected frames

is $[j, j + l]$. As shown in Fig. 13.12, our modified I3D network is able to measure the consistency between consecutive frames.

13.4.4 Experimental Results

We evaluate our proposed C2F-DCNN method on a self-collected video dataset and the Media Forensics Challenge 2018 (MFC18)[4] dataset (Guan et al. 2019).

Our self-collected video dataset is obtained through automatically adding frame duplication manipulation on the 12 raw static camera videos from VIRAT dataset (Oh et al. 2011) and 17 dynamic iPhone 4 videos. The duration of each video is in the range from 47 seconds to 3 minutes. In order to generate test videos with frame duplication, we randomly select frame sequences with the duration 0.5, 1, 2, 5 and 10 s, and then re-insert them into the same source videos. We use the X264 video codec and a frame rate of 30 fps to generate these manipulated videos. Note that we avoid any temporal overlap between the selected original frames and the duplicated frames in all generated videos. Since we have the frame-level ground truth, we can use it for frame-level performance evaluation.

The MFC18 dataset consists of two subsets, Dev dataset and Eval dataset, which we denote as the MFC18-Dev dataset and the MFC18-Eval dataset, respectively. There are 231 videos in the MFC18-Dev dataset and 1036 videos in the MFC18-Eval dataset. The duration of each video is in the range from 2 seconds to 3 minutes. The frame rate for most of the videos is 29–30 fps, while a smaller number of videos are 10 or 60 fps and only five videos in the MFC18-Eval dataset are larger than 240 fps. We opt out these five videos and another two videos which have less than 17

[4] https://www.nist.gov/itl/iad/mig/media-forensics-challenge-2018.

frames from the MFC18-Eval dataset because the input for the I3D network should have at least 17 frames. We use the remaining 1029 videos in the MFC18-Eval dataset to conduct the video-level performance evaluation.

The detection task is to detect whether or not a video has been manipulated with frame duplication manipulation, while the localization task to localize the duplicated frames index. For the measurement metrics, we use the performance measures of area under the ROC curve (AUC) for the detection task, and use the Matthews correlation coefficient

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}$$

for localization evaluation, where TP, FP, TN and FN refer to frames which represent true positive, false positive, true negative and false negative, respectively. See Guan et al. (2019) for further details on the metrics.

13.4.4.1 Frame-Level Analysis on Self-collected Dataset

To better verify the effectiveness of deep learning solution in frame-duplication detection on the self-collected dataset, we consider four baselines: Lin et al.'s method (Guo-Shiang and Jie-Fan 2012) that uses histogram difference as the detection features, Yang et al.'s method (Yang et al. 2016) that is an effective similarity-analysis-based method with SVD features, Ulutas et al.'s method (Ulutas et al. 2017) based on binary features and another method by them (Ulutas et al. 2018) that uses bag-of-words with 130-dimensional SIFT descriptors. Different from our proposed C2F-DCNN method, all of these methods use traditional feature extraction without deep learning.

Note that the manipulated videos are generated by us, hence both selected original frames and duplicated frames are accessible to us. We treat these experiments as a white-box attack and evaluate the performance of frame-to-frame distance measurements.

We run the proposed C2F-DCNN approach and the above-mentioned four state-of-the-art approaches on our self-collected dataset and the results are summarized in Table 13.3. As we can see, due to the X264 codec, the contents of the duplicated frames have been affected so that the detection of a duplicated frame and its corresponding selected frame is very challenging. In this case, our C2F-DCNN method still outperforms the four previous methods.

To help the reader better understand the comparison, we provide a visualization of the normalized distances between the selected frames and the duplicated frames in Fig. 13.13. We can see our C2F-DCNN performs the best for both sample videos, especially with respect to the ability to distinguish the temporal boundary between duplicated frames and non-duplicated frames. All these observations strongly demonstrate the effectiveness of this deep learning approach for frame duplication detection.

Table 13.3 The AUC performance of frame-to-frame distance measurements for frame duplication detection on our self-collected video dataset.(unit: %)

Method	iPhone 4 videos	VIRAT videos
Lin 2012 (Guo-Shiang and Jie-Fan 2012)	80.81	80.75
Yang 2016 (Yang et al. 2016)	73.79	82.13
Ulutas 2017 (Ulutas et al. 2017)	70.46	81.32
Ulutas 2018 (Ulutas et al. 2018)	73.25	69.10
C2F-DCNN	**81.46**	**84.05**

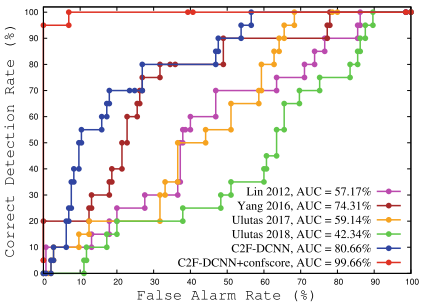

Fig. 13.14 The ROC curves for video-level frame duplication detection on the MFC18-Dev dataset

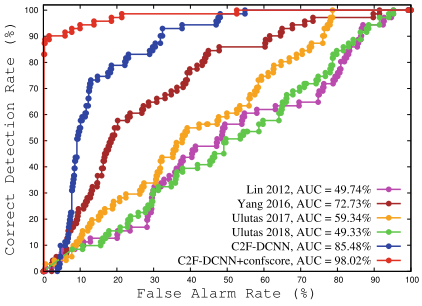

Fig. 13.15 The ROC curves for video-level frame duplication detection on the MFC18-Eval dataset

13.4.4.2 Video-Level Analysis on the MFC18 Dataset

It is worth mentioning that the duplicated videos in the MFC18 dataset usually include multiple manipulations, and this makes the content between the selected original frames and duplicated frames different at times. Therefore, the testing video in both the MFC18-Dev and the MFC18-Eval datasets are very challenging. Since we are not aware of the details about the generation of all the testing videos, we take this dataset as a black-box attack and evaluate its video-level detection and localization performance.

Table 13.4 The MCC metric in [–1.0, 1.0] range for video temporal localization on the MFC18 dataset. Our approach generates the best MCC score, where 1.0 is perfect

Method	MFC18-Dev	MFC18-Eval
Lin 2012 (Guo-Shiang and Jie-Fan 2012)	0.2277	0.1681
Yang 2016 (Yang et al. 2016)	0.1449	0.1548
Ulutas 2017 (Ulutas et al. 2017)	0.2810	0.3147
Ulutas 2018 (Ulutas et al. 2018)	0.0115	0.0391
C2F-DCNN w/ ResNet	0.4618	0.3234
C2F-DCNN w/ C3D	0.6028	0.3488
C2F-DCNN w/ I3D	**0.6612**	**0.3606**

Table 13.5 The video temporal localization performance on the MFC18 dataset. Note $\surd$, $\times$ and $\otimes$ indicate correct cases, incorrect cases and ambiguously incorrect cases, respectively. And #(.) indicates the number of a kind of specific cases

Dataset	#($\surd$)	#($\times$)	#($\otimes$)
MFC18-Dev	14	6	1
MFC18-Eval	33	38	15

We compare the proposed C2F-DCNN method and the above-mentioned four state-of-the-art methods, i.e., Lin 2012 (Guo-Shiang and Jie-Fan 2012), Yang 2016 (Yang et al. 2016), Ulutas 2017 (Ulutas et al. 2017) and Ulutas 2018 (Ulutas et al. 2018) on these two datasets. We use the negative minimum distance (i.e., $-d_{\min}$) as a default video-level scoring method to generate a video-level score for each competing method, including ours. "C2F-DCNN+confscore" denotes our best configuration with C2F-DCNN along with the proposed video-level confidence score defined in Eq. 13.6. In contrast, "C2F-DCNNa" uses only $-d_{\min}$ as the confidence score. The comparative manipulated video detection results are summarized in Figs. 13.14 and 13.15.

A few observations that we would like to point out: (1) C2F-DCNN always outperforms the four previous methods for the video-level frame duplication, with the video-level score as negative minimum distance; (2) with "+conf score", our "C2F-DCNN+confscore" method generates a significant boost in AUC as compared to the baseline score of $-d_{\min}$ and achieves a high correct detection rate at a low false alarm rate; and (3) the proposed "C2F-DCNN+confscore" method achieves very high AUC scores on the two benchmark datasets: 99.66% on MFC18-Dev, and 98.02% on MFC18-Eval.

We also performed a quantified analysis of the temporal localization within a manipulated video with frame duplication. For comparison with the four previous methods, we use the feature distance between any two consecutive frames.

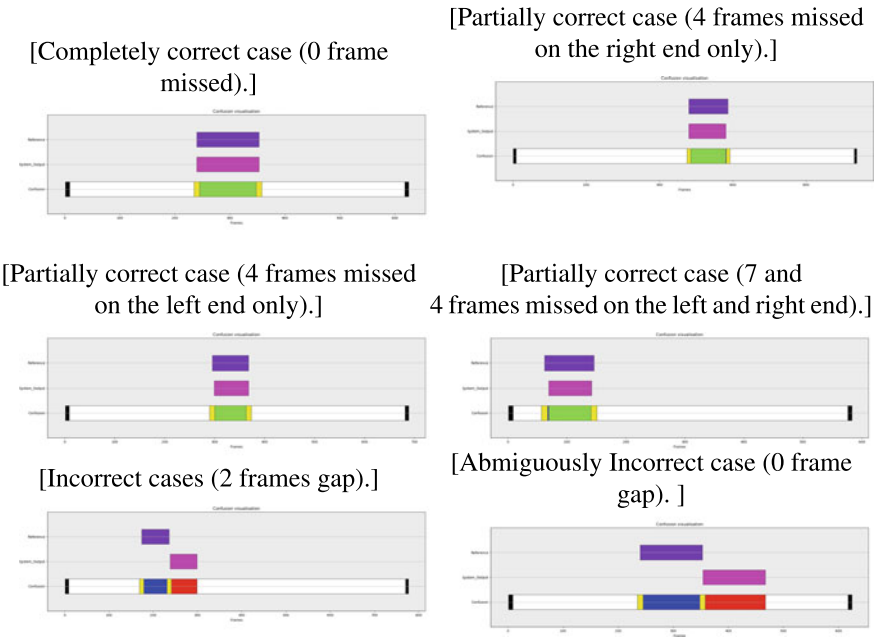

Fig. 13.16 The visualization of confusion bars in video temporal localization. For each subfigure, the top (purple) bar is ground truth indicating duplication, the middle bar (pink) is the system output from the proposed method and the bottom bar is the confusion calculated based on the above the truth and the system output. Note TN, FN, FP, TP and "OptOut" in the confusion are marked in white, blue, red, green and yellow/black, respectively. **a** and **b–d** are correct results, which include completely correct cases and partially correct cases. **e** and **f** show the failure cases

For the proposed C2F-DCNN approach, the best configuration "C2F-DCNN w/ I3D" includes the I3D network as the inconsistency detector. We also provide two baseline variants by replacing the I3D inconsistency detector with a ResNet network feature distance $S_{Res}(i)$ only ("C2F-DCNN w/ ResNet") or the C3D network's scores $S_{C3D}^{drop}(i) - \lambda S_{C3D}^{none}(i)$ from (Chengjiang et al. 2017) ("C2F-DCNN w/ C3D"). The temporal localization results are summarized in Table 13.4, from which we can observe that (1) our deep learning solutions, "C2F-DCNN w/ ResNet", "C2F-DCNN w/ C3D" or "C2F-DCNN w/ I3D" work better than the four previous methods and "C2F-DCNN w/ I3D" performs the best. These observations suggest that 3D convolutional kernel is able to measure the inconsistency between the consecutive frames, and both RGB data stream and optical flow data stream are complementary to further improve the performance.

To better understand the video temporal localization measurement, we plot the confusion bars on the video timeline based on the truth and the corresponding system output under different scenarios, as shown in Fig. 13.16. We would like to emphasize that no algorithm is able to distinguish duplicated frames from selected frames for the ambiguously incorrect cases indicated as $\otimes$ in Table 13.5, because such videos

often break the assumption of temporal consistency and in many cases the duplicated frames are difficult to identify by the naked eye.

13.5 Conclusions and Discussion

We presented a C3D-based network with a confidence score defined with a peak detection step and a scale term for frame dropping detection. The method we proposed in Chengjiang et al. (2017) flexibly explores the underlying spatio-temporal relationship across the one-shot videos. Empirically, it is not only able to identify manipulation of temporal removal type robustly but also to detect the exact location where the frame dropping occurred.

Our future work includes revising frame dropping strategy to be more realistic for training video collection, evaluating an LSTM-based network for quicker runtime, and working on other types of video manipulation detection such as addressing shot boundaries and duplication in looping cases.

Multiple factors cause frame duplication detection and localization becoming more and more challenging in video forgeries. These factors include high frame rates, multiple manipulations (e.g., "SelectCutFrames", "TimeAlterationWarp", "AntiForensicCopyExif", "RemoveCamFingerprintPRNU"[5]) involved before and after, and gaps between the selected frames and the duplicated frames. In particular, zero gap between the selected frames and the duplicated frames renders the manipulation undetectable because the inconsistency which should exist at the end of the duplicated frames does not appear in the video temporal context.

Regarding the runtime, the I3D network for inconsistency detection is the most expensive component in our framework but we only apply it on the candidate frames that are likely to have frame duplication manipulations detected in the coarse-search stage. For each testing video clip with a 16-frame length, it takes about 2 s with our learned I3D network. For a one-minute short video with 30 FPS, it requires less than 5 min to complete the testing throughout all the frame sequences.

The coarse-to-fine deep learning approach is designed for frame duplication detection at both frame-level and video-level, as well as for video temporal localization. This work also included a heuristic strategy to formulate the video-level confidence score, as well as an I3D network-based inconsistency detector to distinguish the duplicated frames from the selected frames. The experimental results have demonstrated the robustness and effectiveness of the method.

Our future work includes continuing to extend multi-stream 3D neural networks for both frame drop, frame duplication and other video manipulation tasks like looping detection, working on frame-rate variations and train on multiple manipulations, investigating the effects of various video codecs on algorithm accuracy.

[5] These manipulation operation are defined in the MFC18 dataset.

Acknowledgements This work was supported by DARPA under Contract No. FA875016-C-0166. Any findings and conclusions or recommendations expressed in this material are solely the responsibility of the authors and does not necessarily represent the official views of DARPA of the U.S. Government. Approved for Public Release, Distribution Unlimited.

References

Aneja J, Deshpande A, Schwing AG (2018) Convolutional image captioning. In: The IEEE conference on computer vision and pattern recognition (CVPR)

Awad G, Le DD, Ngo CW, Nguyen VT, Quénot G, Snoek C, Satoh SI (2010) National institute of informatics, Japan at trecvid 2010. In: TRECVID

Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 4724–4733

Chao J, Jiang X, Sun T (2012) A novel video inter-frame forgery model detection scheme based on optical flow consistency. In: International workshop on digital watermarking, pp 267–281

Chen Y, Li W, Sakaridis C, Dai D, Van Gool L (2018) Domain adaptive faster r-cnn for object detection in the wild. In: The IEEE conference on computer vision and pattern recognition (CVPR)

Christlein V, Riess C, Jordan J, Riess C, Angelopoulou E (2012) An evaluation of popular copy-move forgery detection approaches. arXiv:1208.3665

Conotter V, O'Brien JF, Farid H (2013) Exposing digital forgeries in ballistic motion. IEEE Trans Inf Forensics Secur 7(1):283–296

D'Amiano L, Cozzolino D, Poggi G, Verdoliva L (2019) A patchmatch-based dense-field algorithm for video copy-move detection and localization. IEEE Trans Circuits Syst Video Technol 29(3):669–682

Douze M, Gaidon A, Jegou H, Marszalek M, Schmid C (2008) Inria-lear's video copy detection system. In: TRECVID 2008 workshop participants notebook papers. MD, USA, Gaithersburg

Guan H, Kozak M, Robertson E, Lee Y, Yates AN, Delgado A, Zhou D, Kheyrkhah T, Smith J, Fiscus J (2019) MFC datasets: large-scale benchmark datasets for media forensic challenge evaluation. In: 2019 IEEE winter applications of computer vision workshops (WACVW). IEEE, pp 63–72

Hara K, Kataoka H, Satoh Y (2018) Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and imagenet? In: The IEEE conference on computer vision and pattern recognition (CVPR)

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: The IEEE conference on computer vision and pattern recognition (CVPR)

Huang T, Zhang X, Huang W, Lin L, Weifeng S (2018) A multi-channel approach through fusion of audio for detecting video inter-frame forgery. Comput Secur 77:412–426

Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L (2014) Large-scale video classification with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1725–1732

Kawai Y, Sumiyoshi H, Yagi N (2007) Shot boundary detection at TRECVID 2007. In: TRECVID 2007 workshop participants notebook papers. MD, USA, Gaithersburg

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, vol 25, pp 1097–1105

Law-To J, Buisson O, Gouet-Brunet V, Boujemaa N (2006) Robust voting algorithm based on labels of behavior for video copy detection. In: Proceedings of the 14th ACM international conference on multimedia. ACM, pp 835–844

Lin G-S, Chang J-F (2012) Detection of frame duplication forgery in videos based on spatial and temporal analysis. Int J Pattern Recognit Artif Intell 26(07):1250017

Long C, Basharat A, Hoogs A (2019) A coarse-to-fine deep convolutional neural network framework for frame duplication detection and localization in forged videos. In: IEEE international conference on computer vision and pattern recognition workshop (CVPR-W) on media forensics

Long C, Smith E, Basharat A, Hoogs A (2017) A C3D-based convolutional neural network for frame dropping detection in a single video shot. In: IEEE international conference on computer vision and pattern recognition workshop (CVPR-W) on media forensics

Milani S, Fontani M, Bestagini P, Barni M, Piva A, Tagliasacchi M, Tubaro S (2012) An overview on video forensics. APSIPA Trans Signal Inf Process 1

Mohamadian Z, Pouyan AA (2013) Detection of duplication forgery in digital images in uniform and non-uniform regions. In: 2013 UKSim 15th international conference on computer modelling and simulation (UKSim). IEEE, pp 455–460

Oh S, Hoogs A, Perera A, Cuntoor N, Chen CC, Lee JT, Mukherjee S, Aggarwal JK, Lee H, Davis L, et al (2011) A large-scale benchmark dataset for event recognition in surveillance video. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 3153–3160

Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. In Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems (NIPS), pp 91–99

Smeaton AF, Over P, Doherty AR (2010) Video shot boundary detection: seven years of trecvid activity. CVIU 114(4):411–418

Sowmya KN, Chennamma HR (2015) A survey on video forgery detection. Int J Comput Eng Appl 9(2):17–27

Stock P, Cisse M (2018) Convnets and imagenet beyond accuracy: understanding mistakes and uncovering biases. In: The European conference on computer vision (ECCV)

Tang P, Wang X, Wang A, Yan Y, Liu W, Huang J, Yuille A (2018) Weakly supervised region proposal network and object detection. In: The European conference on computer vision (ECCV)

Thakur MK, Saxena V, Gupta JP (2016) Learning based no reference algorithm for dropped frame identification in uncompressed video, pp 451–459

Thakur MK, Saxena V, Gupta JP (2016) Learning based no reference algorithm for dropped frame identification in uncompressed video. In: Information systems design and intelligent applications: proceedings of third international conference INDIA 2016, vol 3, pp 451–459

Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE international conference on computer vision (ICCV). IEEE, pp 4489–4497

Ulutas G, Ustubioglu B, Ulutas M, Nabiyev V (2017) Frame duplication/mirroring detection method with binary features. IET Image Process 11(5):333–342

Ulutas G, Ustubioglu B, Ulutas M, Nabiyev VV (2018) Frame duplication detection based on bow model. Multimed Syst 1–19

Venugopalan S, Xu H, Donahue J, Rohrbach M, Mooney R, Saenko K (2015) Translating videos to natural language using deep recurrent neural networks. In: North American chapter of the association for computational linguistics—human language technologies (NAACL-HLT)

Wang J, Liu G, Zhang Z, Dai Y, Wang Z (2009) Fast and robust forensics for image region-duplication forgery. Acta Autom Sin 35(12):1488–1495

Wang Q, Li Z, Zhang Z, Ma Q (2014) Video inter-frame forgery identification based on optical flow consistency. Sens Transducers 166(3):229

Wang Q, Li Z, Zhang Z, Ma Q (2014) Video inter-frame forgery identification based on consistency of correlation coefficients of gray values. J Comput Commun 2(04):51

Wang X, Liu Y, Huan X, Wang P, Yang H (2018) Robust copy-move forgery detection using quaternion exponent moments. Pattern Anal Appl 21(2):451–467

Wang W, Farid H (2007) Exposing digital forgeries in video by detecting duplication. In: Proceedings of the 9th workshop on multimedia & security. ACM, pp 35–42

Wolf S (2009) A no reference (NR) and reduced reference (RR) metric for detecting dropped video frames. In: National telecommunications and information administration (NTIA)

Wu Y, Jiang X, Sun T, Wang W (2014) Exposing video inter-frame forgery based on velocity field consistency. In: 2014 IEEE International Conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 2674–2678

Xie S, Sun C, Huang J, Tu Z, Murphy K (2018) Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification. In: The European conference on computer vision (ECCV)

Yang J, Huang T, Lichao S (2016) Using similarity analysis to detect frame duplication forgery in videos. Multimed Tools Appl 75(4):1793–1811

Yang B, Sun X, Guo H, Xia Z, Chen X (2018) A copy-move forgery detection method based on CMFD-SIFT. Multimed Tools Appl 77(1):837–855

Yongjian H, Li C-T, Wang Y, Liu B (2012) An improved fingerprinting algorithm for detection of video frame duplication forgery. Int J Digit Crime Forensics (IJDCF) 4(3):20–32

Yu H, Cheng S, Ni B, Wang M, Zhang J, Yang X (2018) Fine-grained video captioning for sports narrative. In: The IEEE conference on computer vision and pattern recognition (CVPR)

Zhang N, Paluri M, Ranzato MA, Darrell T, Bourdev L (2014) Pose aligned networks for deep attribute modeling. In: CVPR. Panda

Zhao DN, Wang RK, Lu ZM (2018) Inter-frame passive-blind forgery detection for video shot based on similarity analysis. Multimed Tools Appl 1–20

Zheng L, Sun T, Shi YQ (2014) Inter-frame video forgery detection based on block-wise brightness variance descriptor. In: International workshop on digital watermarking, pp 18–30

Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A (2014) Learning deep features for scene recognition using places database. In: NIPS