A C3D-based Convolutional Neural Network for Frame Dropping Detection in a Single Video Shot

Chengjiang Long Eric Smith Arslan Basharat Anthony Hoogs
Kitware Inc.
28 Corporate Drive, Clifton Park, NY 12065
{chengjiang.long, eric.smith, arslan.basharat, anthony.hoogs}@kitware.com

Abstract

Frame dropping is a type of video manipulation where consecutive frames are deleted to omit content from the original video. Automatically detecting dropped frames across a large archive of videos while maintaining a low false alarm rate is a challenging task in digital video forensics. We propose a new approach for forensic analysis by exploiting the local spatio-temporal relationships within a portion of a video to robustly detect frame removals. In this paper, we propose to adapt the Convolutional 3D Neural Network (C3D) for frame drop detection. In order to further suppress the errors due by the network, we produce a refined video-level confidence score and demonstrate that it is superior to the raw output scores from the network. We conduct experiments on two challenging video datasets containing rapid camera motion and zoom changes. The experimental results clearly demonstrate the efficacy of the proposed approach.

1. Introduction

Digital video forgery [6] 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.

In this paper, we focus on the problem of video frame drop detection in a given, possibly manipulated, video without the original video. As illustrated in Figure 1, we define a frame drop to be a removal of any number of consecutive frames from a within a video shot[1]. In this paper we consider only videos with a single shot to avoid the confusion between frame drops and shot breaks. Today, single shot videos are prevalent from a variety of sources like mobile phones, car dashboard cameras, or body worn cameras.

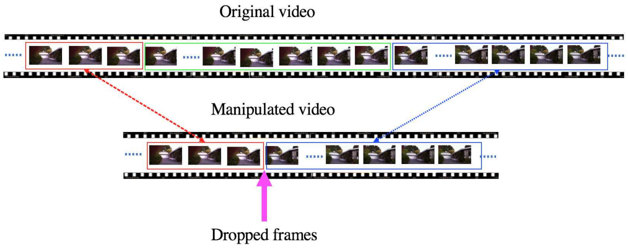


Figure 1: The illustration of frame dropping detection challenge. Assuming that there are three consecutive frame sequences (marked in red, green and blue, respectively) in an original video, the manipulated video is obtained after removing the green frame sequence. Our goal is to identify the location of the frame drop at the end of the red frame sequence and the beginning of the blue frame sequence.

To the best of our knowledge, only a small amount of recent work [8] 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 [13] 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* [7] proposed a SVM-based method to classify tampered or non-tampered videos. In this paper, we explore the authentication [2, 12] 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 an 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.

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

We shall emphasize that we still can use the 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 [13] and optical flow [1, 11] 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 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 [3]. The features computed by these algorithms have been used for image matching/classification [14, 16]. In this paper we evaluate approaches using these features for dropped frame detection using two to three frames. However, these image-based deep features still lack modelling the motion effectively.

Inspired by Tran *et al*'s C3D network [9], 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 Figure 2. As we can observed, there are three aspects distinguish our proposed C3D-based network from Tran *et al*'s work. (1) Our task is to check whether there exist frames dropped between the 8-th and 9-th frames, 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" respectively between the 8-th and 9-th frames; (3) unlike most approaches that use the output scores from the network as confidence score directly, we define confidence score with a peak detection trick and a scale term based on the output score curves; and (4) such a network is able not only to predict whether the video has frame dropping, but also to detect the exact location where the frame dropping occurs.

To summarize, our contributions in this paper are:

- We propose a 3D convolutional network for frame dropping detection, and the confidence score is defined with a peak detection trick and a scale term based on the output score curves. It is able to identify whether frame dropping exists and even determine the exact location of frame dropping without any information of the reference/original video.
- For performance comparison, we also compare to 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.

2. Related work

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 deleting. Wang *et al* proposed a SVM method [11] 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 [1] 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* [10] also extracted the consistency of correlation coefficients of gray values as distinguishing features to classify original videos and forgeries. Zheng *et al* [15] proposed a novel feature called block-wise brightness variance descriptor (BBVD) for fast detecting video inter-frame forgery. Different from these inter-frame forgery identification, our proposed C3D-based network 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 shot boundary detection problem [4]. The task of shot boundary detection [5] 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 short 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.

3. Algorithms

To clarify, there is little work exploring on frame dropping detection problem without reference or original video. Therefore, we first develop a series of baselines including cue-based and learning-based methods, and then introduce our proposed C3D-based CNN.

3.1. Baselines

We implement three different cue-based baseline algorithms from the literature, *i.e.*, (1) color histogram, (2) optical flow [11, 1], (3) motion energy [13] as follows:

- **Color histogram**. We calculate the histograms on all R, G and B three channels. Whether there are frames

Method	Brief description	Learning?
Color histogram	RGB 3 channel histograms + L2 distance.	No
Optical flow	The optic flow [11, 1] with Lucas-Kanade method + L2 distance.	No
Motion energy	Based on temporal information difference [13] sequence.	No
SVM	770-D feature vector (3x256-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 [3] Network	Alexnet-variant network architecture.	Yes
C3D-based Network	C3D-variant network architecture + confidence score.	Yes

Table 1: A list of competing algorithms. The first three algorithms are cue-based with out any training work. The rest are learned-based algorithms including the traditional SVM, the popular CNNs and our proposed method (the last one) in this paper.

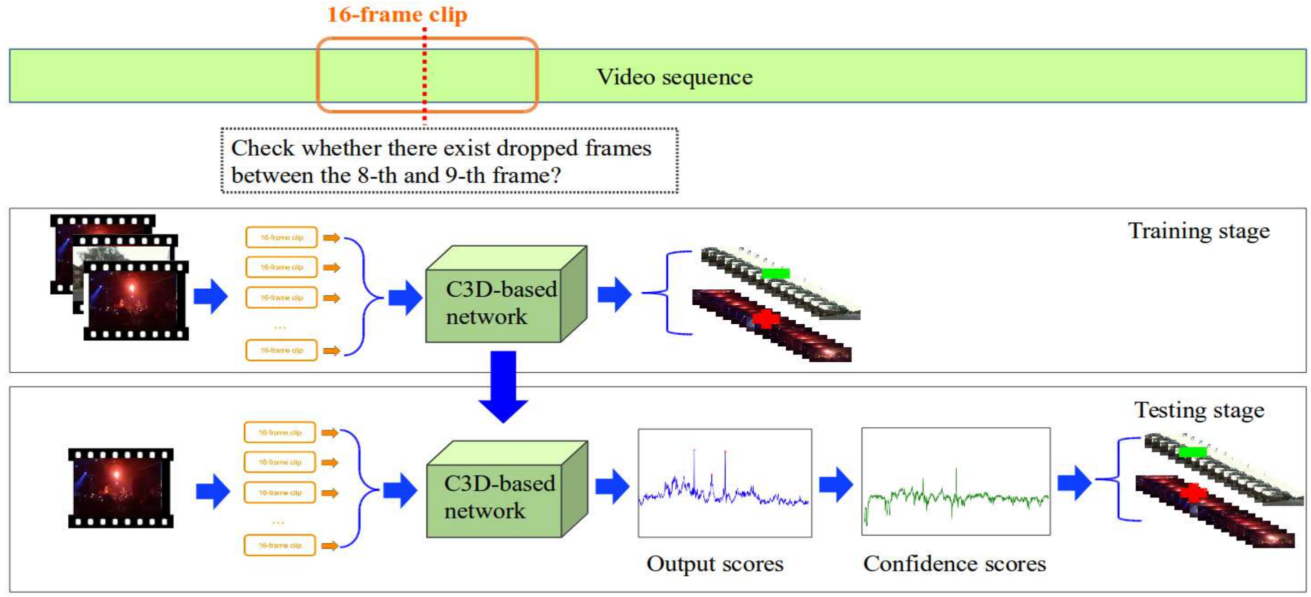


Figure 2: The pipeline of the proposed C3D-based method. At the training stage, the C3D-based network takes 16-frame video clips extracted from the video dataset as input, and produces two outputs, *i.e.*, "frame drop" (indicated with "+") or "no frame drop" (indicated with "-"). At testing stage, we 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 trick 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 is consisted of 66 million parameters with 3x3x3 filter size at all convolutional layers.

dropped between the two consecutive frames is detected by thresholding the score calculated by the L2 distances based on the color histograms of these adjacent two frames.

- **Optical flow**. We calculate the optical flow [11, 1] from the adjacent two frames by Lucas-Kanade method. Whether there exist frames dropped between the 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 [13], *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 above algorithm compares two consecutive frames and estimates whether there are missing frames between them. We also develop 4 learning-based baseline algorithms as follows:

- **SVM**. We train a 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 2-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 Network consisted of three the Alexnets [3] 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 grey-scale and put into the RGB channels.

To facilitate the comparison of the competing algorithms, we summarize the above descriptions in Table 1.

3.2. Proposed method

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 Figure 2. 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 8-th and 9-th frames. Lastly, we further refine the per-frame network output scores into a confidence score using peak detection and temporal sclaing to further suppress the noisy detections. With the refined confidence scores we are able not only to identify whether the video has a frame drops, but also to localize them by applying the network to the video in a sliding window fashion.

3.2.1 Data preparation

To obtain the training data, we download 2,394 iPhone 4 videos from the World Dataset via the Medifor RankOne Browser[2]. We kept the videos such that all videos were of length 1-3 minutes. We ended up with 314 videos, of which we randomly selected 264 video for training, and adopted 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, 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 0.5s, 1s, 2s, 5s, and 10s as five different frame drop durations.

We gather the videos with the above 5 drop durations together to train a general C3D-based network for frame drop detection.

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 206000-th iteration as the learned model for testing, which achieves about 98.2% validation accuracy.

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 a the confidence scores using a 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 for 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 $i-th$ frame as the center (denoted as $W(i)$). If the number

[2] The URL for the Medifor RankOne Browser is https://medifor.rankone.io.

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 $i-$th frame as

$$f_{conf}(i) = \begin{cases} f(i) - \lambda\Delta(i) & \text{when } n_p < 2 \\ \frac{f(i)}{n_p} - \lambda\Delta(i) & \text{otherwise} \end{cases} \tag{1}$$

where

$$\Delta(i) = \underset{k \in W(i)}{\text{median}}\, f(k) - \min_{k \in W(i)} f(k) \tag{2}$$

$$W(i) = \{i - \frac{w}{2}, \ldots, i + \frac{w}{2}\}. \tag{3}$$

Note that λ in Equation 1 is a parameter to control how much the scale term affects the confidence score, and w in Equation 3 indicates the width of the time window.

For testing per-frame, say $i-$th frame, we first form a 16-frame video clip and set the $i-$th frame to be the $8-$th 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 $i-$th frame and the $(i+1)-$th frame. For testing on per video, we take it as a binary classification and confidence measure per video. To make it simple, 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 in the video. Otherwise, the video is predicted without any temporal removal. The results reported in this paper are without any $Threshold$ as we are reporting the ROC curves.

4. Experiments

We conducted the experiments on a Linux machine with Intel(R) Xeon(R) CPU E5-2687 0 @ 3.10GHz, 32 GB system memory and Graphical card NVIDIA GTX 1080 (Pascal). We report our results as the ROC curves based on the output score $f_{conf}(i)$ and accuracy as metrics. We present ROC curves with with false positive rate as well as false alarm rate per minute to provide a to demonstrate the level of usefulfness for a user that might have to adjudicate each detection reported by the algorithm. We present 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[3] and the Nimble Challenge 2017 (Development 2 Beta 1) dataset[4].

4.1. Experiments on the 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 videos with frame drops of 0.5, 1, 2, 5, or 10 seconds intervals at random locations. For each video and for 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.

4.1.1 Performance Comparison

For each drop duration, we run all the competing algorithms in Table 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 Figure 3.

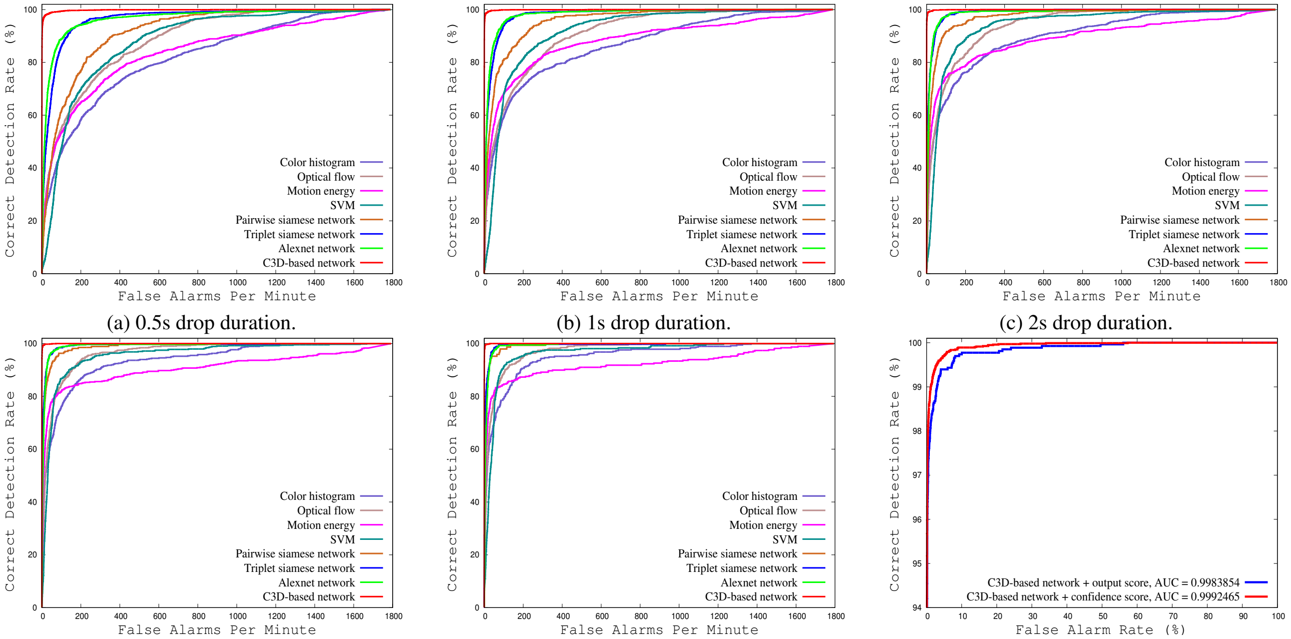
One can note that: (1) the traditional SVM outperforms the three simple cue-based algorithms; (2) the four convolution neural networks algorithms perform much better than the traditional SVM and all the cue-based algorithms; (3) among all the CNN-based networks, both the Triplet-Siamese network and the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

To better understand our proposed C3D-based network, we provide more experimental details in Table 2. Obviously, 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 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 Figure 3 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

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

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



(a) 0.5s drop duration.

(b) 1s drop duration.

(c) 2s drop duration.

(d) 5s drop duration.

(e) 10s drop duration.

(f) Overall ROC with all drop durations.

Figure 3: 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 that are shown separately in (a-e).

$Duration$	$\#_{pos} : \#_{neg}$	Acc_{pos}	Acc_{neg}	Acc
0.5s	2816:416633	98.40	98.15	98.16
1s	2333:390019	99.49	98.18	98.18
2s	1991:342845	99.57	98.11	98.12
5s	1225:282355	99.70	98.17	98.18
10s	770:239210	100.00	98.12	98.13

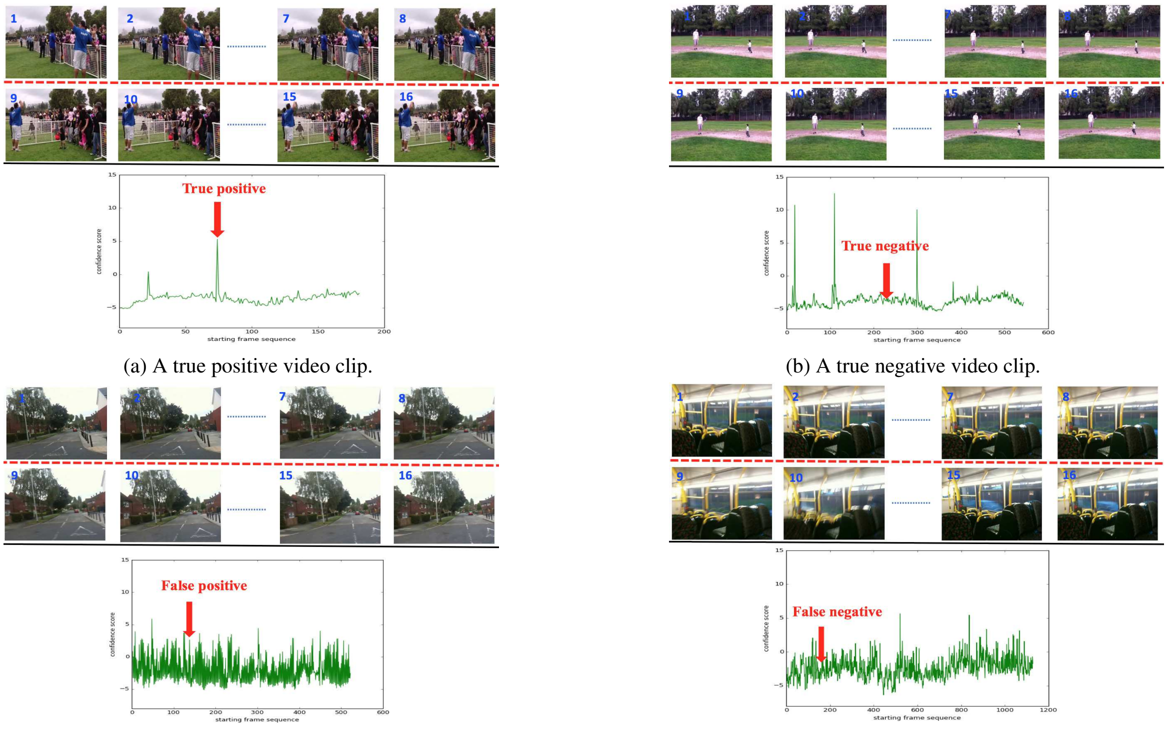
Table 2: The detailed results of our proposed C3D-based network. $\#_{pos}$ and $\#_{neg}$ are the number instances for the positive and the negative testing 16-frame video clips, respectively. The Acc_{pos} and Acc_{neg} are the corresponding accuracy. Acc is the total accuracy. All the accuracies use the unit % and use zero as the threshold value.

we can see in Figure 3f, using output score from the C3D-based network straightly, we still can achieve a 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 any 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 strongly demonstrates the power and the generalization ability of our trained network. Although using output score directly achieves a very good AUC, using the confidence score defined in Equation 1 can still improve the AUC from 0.9983854 to 0.9992465. This strongly demonstrates the effectiveness of our confidence score defined with such a peak detection trick and a scale term.

4.1.2 Visualization

We visualize both success cases and failure cases in our proposed C3D-based network, as shown in Figure 4. Looking at the successful cases in Figure 4a, "frame drops" is identified correctly in the 16-frame video clip because a man stands at one side in the 8−th frame and move to another side suddenly in the 9−th frame, and the video clip in Figure 4b is predicted as "no frame drops" correctly since a child follows his father in all 16 frames and the 8-th frame and the 9-th frame are consistent with each other.

Regarding the failures cases, as shown in Figure 4c(c), there is no frame drop but it is still identified as "frame drop" between the 8−th frame and the 9−th 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 8−th frame and the 9−th frame in the video clip shown in Figure 4c(d), since the scene inside the bus has almost no visible changes between these two frames.



(a) A true positive video clip.

(b) A true negative video clip.

(c) A false positive video clip.

(d) A false negative video clip.

Figure 4: 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 YFCC100m dataset. The red dashed line indicates the location between the 8-th frame and the 9-th frame where we test for a frame drop. The red arrows point to the frame on the confidence socre plots.

4.1.3 Runtime

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

4.2. Experiments on the 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 continue to conduct experiments on the Nimble Challenge 2017 dataset especially NC2017-Dev2Beta1 version, in which there are 209 probe videos with various video manipulations. Among these videos, there are 6 videos manipulated with “TemporalRemove”, which is regarded as “frame dropping” identically. 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 video level. In this experiment, the parameters are set as $w = 500$, $\lambda = 1.25$.

We firstly plot the output scores from the C3D-based network and the confidence score each of the 6 videos labeled with “TemporalRemove” in Figure 7. It is clear that the video named “d3c6bf5f224070f1df74a63c232e360b.mp4” has a lowest confidence score smaller than zero.

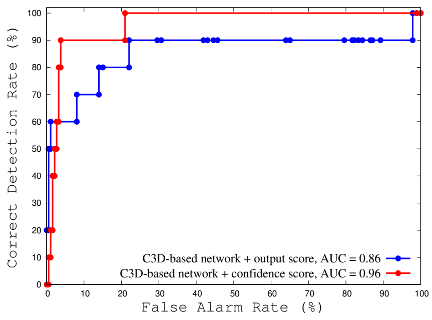


Figure 5: The ROC curve of our proposed C3D-based network on the Nimble Challenge 2017 dataset.

To explain such a case, we further check the content of the video, as shown in Figure 6. As we can observe, this video is even really hard for us to identify it as “TemporalRemoval” since it 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

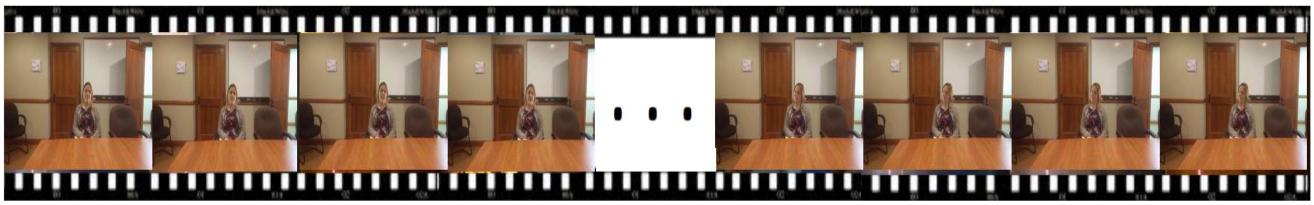


Figure 6: The entire frame sequence of the 34-second video “d3c6bf5f224070f1df74a63c232e360b.mp4”, which has 1047 frames and was captured by a static camera. We observe that only the lady’s mouth and head are taking very slight change across the video from the beginning to the end.

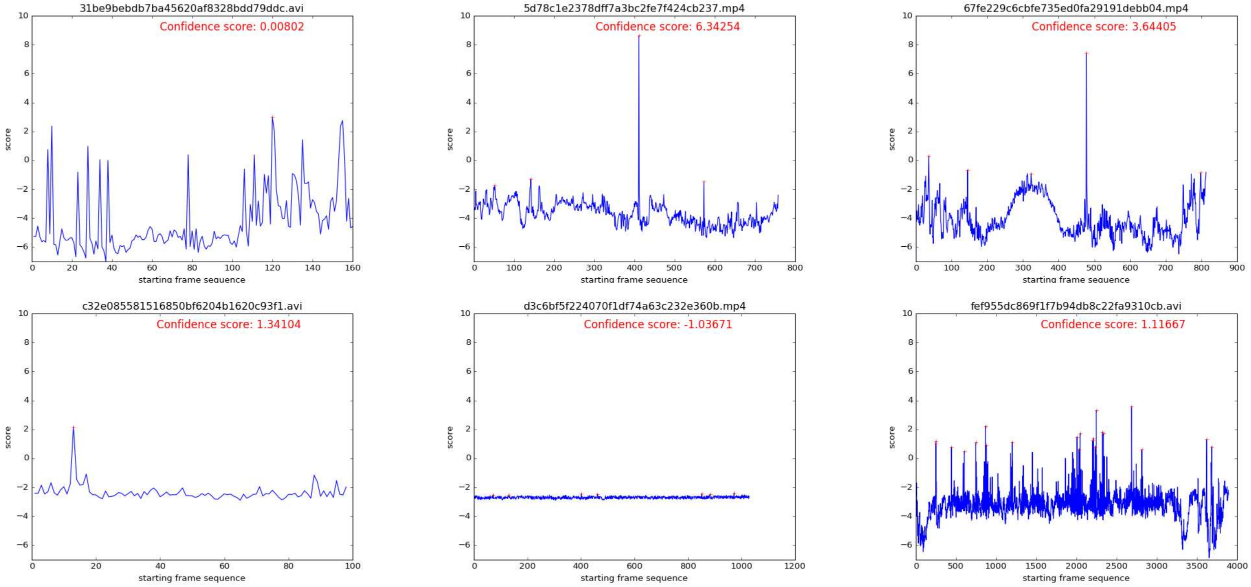


Figure 7: The illustration of output scores from the C3D-based network and their confidence scores for six videos labeled with “TemporalRemove” from the Nimble Challenge 2017 dataset. The blue curve is the output score, the red “+” marks the detected peaks, and the red confidence score is used to determine whether the video can be predicted as a video with “frame drops”.

trained purely on iPhone videos, our training network was biased 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 Figure 5. 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 fact to explain 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 trick and the scale term to penalize multiple peaks occurring too close and large scales is able to significantly reduces the false alarms. Obviously, such a significant improvement by 0.11 AUC strongly demonstrates the effectiveness of our proposed method.

5. Conclusion

We present a C3D-based network with confidence score defined with a peak detection trick and a scale term for frame dropping detection. Our proposed method flexibly explore the underlying spatio-temporal relationship across the one-shot videos. Experimentally it is able not only 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 a Long Short-term Memory (LSTM) based network for quicker run time, and working on other types of video manipulation detection such as addressing shot boundaries and duplication in looping cases.

Acknowledgement

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

[1] J. Chao, X. Jiang, and T. Sun. A novel video inter-frame forgery model detection scheme based on optical flow consistency. In *International Workshop on Digital Watermarking*, pages 267–281, 2012.

[2] V. Conotter, J. F. O'Brien, and H. Farid. Exposing digital forgeries in ballistic motion. *IEEE Transactions on Information Forensics and Security*, 7(1):283–296, 2012.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012.

[4] A. F. Smeaton, P. Over, and A. R. Doherty. Video shot boundary detection: Seven years of trecvid activity. *CVIU*, 114(4):411–418, 2010.

[5] A. F. Smeaton, P. Over, and A. R. Doherty. Video shot boundary detection: Seven years of trecvid activity. *CVIU*, 114(4):411–418, 2010.

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

[7] M. K. Thakur, V. Saxena, and J. Gupta. Learning based no reference algorithm for dropped frame identification in uncompressed video. pages 451–459, 2016.

[8] M. K. Thakur, V. Saxena, and J. P. Gupta. 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, Volume 3*, pages 451–459, 2016.

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

[10] Q. Wang, Z. Li, Z. Zhang, and Q. Ma. Video inter-frame forgery identification based on consistency of correlation coefficients of gray values. *Journal of Computer and Communications*, 2(04):51, 2014.

[11] Q. Wang, Z. Li, Z. Zhang, and Q. Ma. Video inter-frame forgery identification based on optical flow consistency. *Sensors & Transducers*, 166(3):229, 2014.

[12] W. Wang and H. Farid. Exposing digital forgeries in video by detecting duplication. In *Proceedings of the 9th workshop on Multimedia & security*, pages 35–42. ACM, 2007.

[13] S. Wolf. A no reference (nr) and reduced reference (rr) metric for detecting dropped video frames. In *National Telecommunications and Information Administration (NTIA)*, 2009.

[14] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *CVPR*, 2014.

[15] L. Zheng, T. Sun, and Y.-Q. Shi. Inter-frame video forgery detection based on block-wise brightness variance descriptor. In *International Workshop on Digital Watermarking*, pages 18–30, 2014.

[16] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.