symmetry

MDPI

Article

Deep Learning for Detection of Object-Based Forgery in Advanced Video

Ye Yao [1], Yunqing Shi [2], Shaowei Weng [3,*] and Bo Guan [4,*]

1 School of CyberSpace, Hangzhou Dianzi University, Hangzhou 310018, China; yyaoprivate@gmail.com
2 Department of ECE, New Jersey Institute of Technology, Newark, NJ 07102, USA; shi@njit.edu
3 School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China
4 Department of Regional Cooperation, Ningbo University of Technology, Ningbo 315211, China
* Correspondence: wengsw@gdut.edu.cn (S.W.); guanbo@zjinfo.gov.cn (B.G.)

Received: 26 October 2017; Accepted: 21 December 2017; Published: 26 December 2017

Abstract: Passive video forensics has drawn much attention in recent years. However, research on detection of object-based forgery, especially for forged video encoded with advanced codec frameworks, is still a great challenge. In this paper, we propose a deep learning-based approach to detect object-based forgery in the advanced video. The presented deep learning approach utilizes a convolutional neural network (CNN) to automatically extract high-dimension features from the input image patches. Different from the traditional CNN models used in computer vision domain, we let video frames go through three preprocessing layers before being fed into our CNN model. They include a frame absolute difference layer to cut down temporal redundancy between video frames, a max pooling layer to reduce computational complexity of image convolution, and a high-pass filter layer to enhance the residual signal left by video forgery. In addition, an asymmetric data augmentation strategy has been established to get a similar number of positive and negative image patches before the training. The experiments have demonstrated that the proposed CNN-based model with the preprocessing layers has achieved excellent results.

Keywords: deep learning approach; convolutional neural network; video object forgery detection; forgery detection and temporal localization

1. Introduction

Due to the rapid development of digital video technology, editing or tampering a video sequence becomes much easier than before. Everyone can remove an object in a video sequence with the aid of powerful video editing software, e.g., Adobe Premiere, Adobe After Effects, and Apple Final Cut Pro. The videos with alternated objects spreading over the Internet often interfere with our understanding of the video content, and lead to a serious social security event [1]. In recent years, more and more researchers focus on the study of video tampering detection. Object forgery detection has become a new topic in the research field of digital video passive forensics [2].

Object forgery in video is a common video tampering method by means of adding new objects to a video sequence or removing existing ones [3,4]. In contrast to the image copy-move forensics approaches [5–7], video object tamper detection is a more challenging task. If we use image forensics algorithms to detect video tampering, the computational cost will be unacceptable. As a result, the methods for image forensics cannot be applied straightforwardly to video forensics. The temporal correlation between video frames should be considered to reduce the complexity of video forensics.

For this purpose, several video forensic algorithms have been proposed in recent years [8]. Some of these methods analyse pixel-similarity between different video frames. Bestagini et al. [9] presented two algorithms to detect the image-based attack and the video-based attack based on exploiting a correlation analysis for image pixels and image blocks. Lin and Tsay [10] presented a passive approach for effective

detection and localization of region-level forgery from video sequences based on spatio-temporal coherence analysis. Other methods extract specific statistical features, then do classification with machine learning algorithms. In [11], several statistical features have been proposed for classification through SVM (support vector machine) algorithms. In [12], SIFT features are extracted from the video frames, then K-NN matching is used to find out spatial copy-move forgery. Chen et al. [3,4] has adopted the 548 dimensional CC-PEV image steganalysis features to detect the alteration inside the motion residuals of the video frames, then use ensemble classifier to locate the forged video segments in the forged videos. However, all the above methods have depended on manually designed features from tampered video sequences and pristine video sequences.

In recent years, deep learning-based techniques, such as convolutional neural network(CNN), have achieved a great success in the field of computer vision. Deep neural networks have the ability to extract complex high dimensional features and make efficient representations. More recently, deep learning-based approach has been used in many new fields, such as camera model identification [13,14], steganalysis [15], image recapture forensics [16], image manipulation detection [17], image copy-move forgery detection [18], and so on.

In this paper, we propose a new video object tamper detection approach based on deep learning framework. The main contributions are described as follows: (1) We propose a CNN-based model with five layers to automatically learn high-dimension features from the input image patches; (2) Different from the traditional CNN models in the field of computer vision, we let video frames go through three preprocessing layers before feeding our CNN model. In the first layer, the absolute difference of consecutive frames can be calculated so as to cut down temporal redundancy between video frames and reveal the trace of tampering operation; (3) The second layer is a max pooling layer to reduce computation complexity of image convolutional; (4) The third layer is a high-pass filter layer to strengthen the residual signal left by video forgery; (5) After the first layer and before the second layer, we adopt an asymmetric data augmentation strategy to get a similar number of positive and negative image patches. This is a data augmentation method based on video frame clipping for neural network training to avoid overfitting and improve the network generalization capability.

2. Proposed Method

The proposed method is composed of two main steps, i.e., video sequence preprocessing and network model training. In the first step, an absolute difference algorithm is applied to the input video sequence, then the output difference frames are clipped to image patches by means of asymmetric data augmentation strategy. In this way, the input video sequence is converted to image patches. We label these image patches as positive and negative samples, which constitute the training data set. In the second step, the training data set is processed with a max pooling layer and a high pass filter, and then the output is used to train a five-layer CNN-based model. In this section, we discuss the two steps of our proposed method in details.

2.1. Video Sequence Preprocessing

The video sequence consists of a number of consecutive image frames. These image frames are very similar to each other. The only difference between adjacent frames is the status of moving video objects. Object-based forgery means that video objects are copied and moved elsewhere in the video sequence or removed by manipulation of inpainting. These tampering operations leave some perceptible traces inevitably. In order to detect these tampering traces, we propose to compute the absolute difference between consecutive frames to reduce the temporal redundancy, and then clip the residual sequence of absolute difference to residual image patches. Therefore, object-based forgery detection in video sequence can be regarded as forensics of the modification in residual image patches.

2.1.1. Absolute Difference of Consecutive Frames

In order to input video data into the deep neural network model, the video frames are converted to motion residual images through applying the proposed absolute difference algorithms. The proposed algorithms consist of the following steps: converting each frame of the video sequence into a gray-scale image firstly, and then starting from the second grayscale image, subtracting its previous grayscale image from the current ones, and finally taking the absolute value of the subtracted result to obtain absolute difference images, which represent the motion residual between consecutive video frames.

Denote the input video sequence of decoded video frames of length N as

$$S = \{F_1, F_2, ..., F_i, ..., F_N\}, i \in \{1, \ldots, N\} \tag{1}$$

where F_i represents the ith decoded video frame.

Since the decoded video frame is decompressed from advanced video with advanced encoding standards, it has a color space of RGB with R, G, B components. In order to reduce the computational complexity, we convert the decoded video frame into a grayscale image and then perform subtraction and absolute operations. Finally, we can have the absolute difference image D_j as

$$D_j = abs(Gray(F_j) - Gray(F_{j-1})), \quad j \in \{2, \ldots, N\} \tag{2}$$

where the function of $Gray()$ represent color space converting from RGB to gray-scale, $abs()$ is the absolute operation for the difference of two adjacent gray-scale image.

The subtraction operation starts from the second video frame. Note that according to Formula (2), the pixel value outputed by $Gray()$ is limited to $[0, 255]$. Therefore, the resulting D_j can be regarded as an 8-bit gray-scale absolute difference image, which represents the motion residual between consecutive video frames.

2.1.2. Asymmetric Data Augmentation

A large number of training data can be a great help to avoid overfitting and improve the network model generalization capability. Data augmentation [19] provides a method for increasing the amount of training data available for machine learning. It has been extensively adopted in the area of deep learning research for computer vision [19]. In this paper, we present an asymmetric data augmentation strategy to generate more image patches. These image patches are labeled as positive or negative samples for training.

In order to get more training data, the gray-scale frame absolute difference image D_j, which was calculated from Formula (2), is clipped into several image patches. All of the clip operations can achieve label-preserving. Namely, to prepare the positive samples (tampered image patches), we can draw image patches from tampered frames. In a similar way, we can get negative samples (un-tampered image patches) from pristine frames. All video frames in the pristine video sequence are pristine frames, but there are pristine frames and tampered frames in tampered video sequences. Therefore, the number of pristine frames is far more than the tampered ones. In light of this, we draw more image patches in each tampered frames than we draw in pristine frames. This clipping method for image patches is named as an asymmetric data augmentation strategy and shown in Figure 1.

As shown in Figure 1, we draw three image patches from the pristine frame (Figure 1a) with a suitable stride size, and label these image patches as negative samples. The three image patches are respectively located on the left, right and central position of the pristine frame. In the tampered frame (Figure 1b), the moving pedestrian has been removed from the scene. The tampered region is marked with a rectangle R, and point C is the central point of rectangle R. We draw M image patches from the right tampered frame with a stride size 10, and label these image patches as positive samples. To ensure all of the M image patch are positive samples, the point C must be contained in each image patches. In other words, the number of positive samples M is limited to an appropriate

value. Therefore, the quantity of positive samples and the quantity of negative samples are similar by using the proposed asymmetric data augmentation strategy.

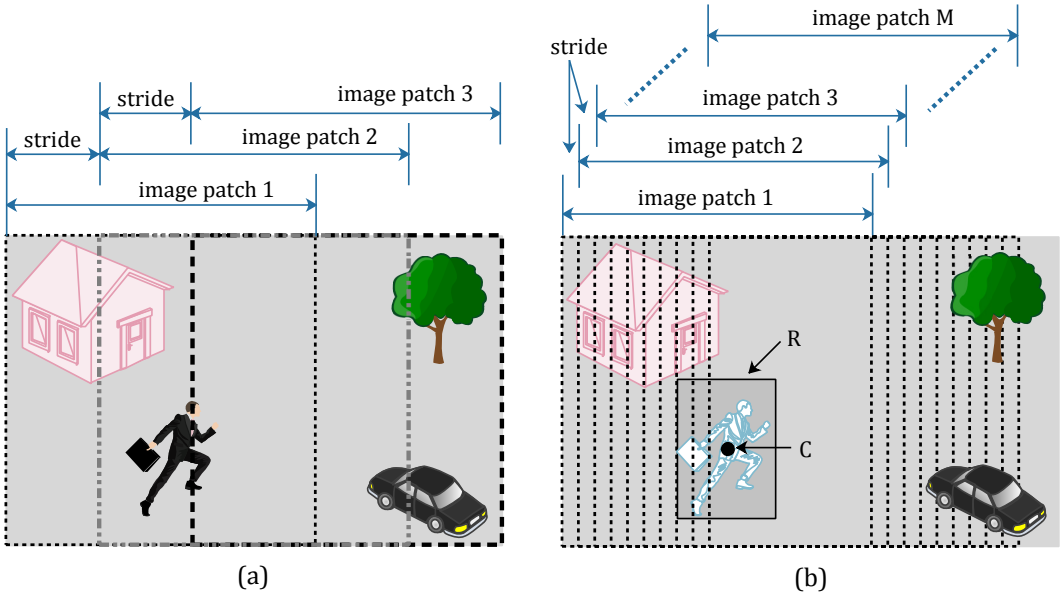

Figure 1. Asymmetric data augmentation strategy. (**a**) Draw three image patches from the pristine frame; (**b**) Draw *M* image patches from the tampered frame.

2.2. Network Architecture

The proposed network architecture is shown in Figure 2. There are a max pooling layer and a high-pass filter layer at the very beginning of the proposed architecture. The proposed network is a five-layer CNN-based model.

2.2.1. Max Pooling

Max pooling is such a subsampling scheme that the maximum value of the input block is returned. It is widely used in some deep learning networks. At the front of the proposed network, a max pooling layer is not only used to reduce the resolution of the input image patches but also used to make the network robust to the variations on motion residual values of the frame absolute difference image.

The input image patches of the CNN-based model are image blocks of 2-D array with size $1 \times (720 \times 720)$ (1 represent channel number of gray-scale). With a window size of 3×3 and a stride size of 3, the resolution of the image patch is reduced from 720×720 to 240×240 after the max pooling layer.

2.2.2. High Pass Filter

Qian et al. [20] presented a predefined high pass filter, which was also defined as a SQUARE5 $\times$ 5 residual class in [21], for image steganalysis to enhance the weak steganography signal and suppress the impact caused by image content. This high-pass filter is a 5×5 shift-invariant convolution kernel. The value of this kernel keeps fixed during training. This filtering has been applied to deep learning-based steganalysis [15] as well as to deep learning-based camera model identification [14], and has achieved good performance.

In this paper, we use the high-pass filter in our video tamper detection to strengthen the residual signal at the frame absolute difference image, and to reduce the impact caused by video object motion between video frames. The fixed high-pass filter is applied to the input image patches, then the filtered image patches are fed to our CNN-based model.

2.2.3. CNN Based Model

The proposed CNN-based model is illustrated in Figure 2. This CNN-based model consists of 5 convolution layers. Each convolution layer is followed by Batch Normalization (BN) [22], Rectified Linear Units (ReLU) [23], and Average Pooling. At the end of proposed network, a fully connected layer and a softmax layer are used to convert 128-D feature vectors to 2-D probabilities, then the classification results are outputted based on the 2-D probabilities.

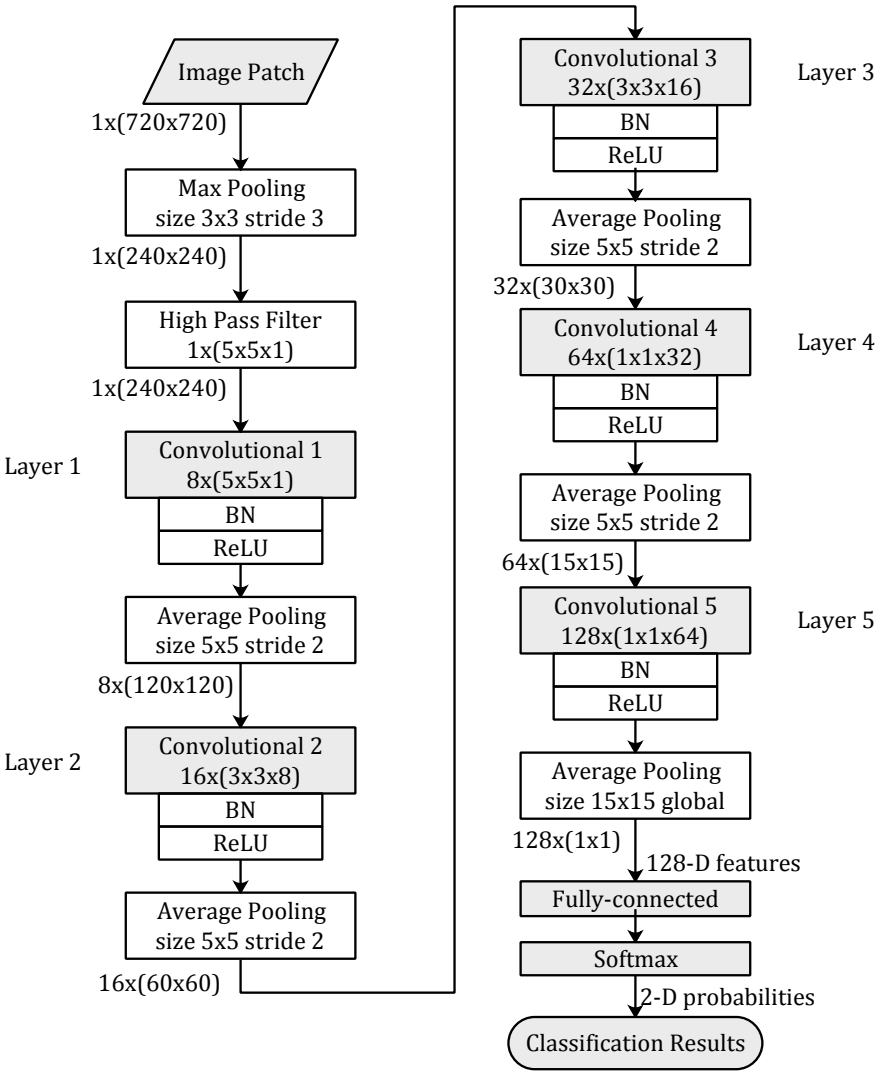

Figure 2. The network architecture of proposed method. Layer functions and parameters are displayed in the boxes. Kernels sizes of convolution in each layers are described in $number\_of\_kernels \times (width \times height \times number\_of\_input)$. Sizes of feature maps between different layers are described in $number\_of\_feature\_maps \times (width \times height)$. To keep the shape of image patches, padding is applied in each layer.

The kernel's sizes in the five convolution layers are 5×5, 3×3, 3×3, 1×1, 1×1, respectively, and the corresponding amounts of feature maps are 8, 16, 32, 64, 128, respectively. The size of feature maps are 240×240, 120×120, 60×60, 30×30, 15×15, respectively. The window sizes of each average pooling layers are 5×5 and stride size is 2, except the last average pooling layer with a global window size of 15×15.

3. Experimental

3.1. Dataset

We test our deep learning-based method on SYSU-OBJFORG data set [3]. SYSU-OBJFORG is the largest object-based forged video data set according to the report in [3]. It consists of 100 pristine video sequences and 100 forged video sequences. The pristine video sequences are directly cut from some primitive video sequences obtained with several static video surveillance cameras without any type of tampering. Every forged video sequence is tampered from one of the corresponding pristine video sequence by means of changing moving objects in the scene. All video sequences are of 25 frames/s, 1280×720 H.264/MPEG-4 encoded video sequences with a bitrate of 3 Mbit/s.

In our experiment, 50 pairs of video sequences, which consist of 50 pristine video sequences and 50 tampered video sequences, are used for training and validation. The other 50 pairs are set aside for testing. We divide the 50 pairs in training and validation data set into five non-overlapping parts. At training stage, one of the five parts is used for validation, and the rest parts are used for training. The training stage is conducted five times based on different validation data set and training data set at each time. After five times of training stage, we obtain five trained CNN-based models with different weights and parameters.

In the testing stage, the 50 pairs of testing video sequences are converted to image patches by means described in Figure 1a firstly. Then all of the image patches are fed to each of the five trained CNN-based models. As a result, we can get five probabilities for each test image patch. The five probabilities are averaged to get a classification for each test image patch. By means of averaging the five probabilities, we can get more accurate and more robust classification results.

3.2. Experimental Setup

The proposed CNN-based model is implemented based on the Caffe deep learning framework [24] and executed on a NVIDIA GeForce GTX 1080ti GPU. Stochastic Gradient Descent is used to optimize our CNN-based model. We set the parameters of *momentum* to 0.9, and *weight_decay* to 0.0005. The initial learning rate is set to 0.001. The learning rate update policy is set to *inv* with the *gamma* value of 0.0001 and the *power* value of 0.75. We set the batch size for training to 64, namely 64 image patches (as positive and negative samples) are input for each iteration. After 120,000 iterations, we can obtain the trained CNN-based model with trained weights and parameters for testing.

In order to verify the performance of the trained CNN-based model on the testing data set, all of the testing video sequences need to be preprocessed. The preprocessing procedure for testing data set is similar to the procedure for training data set. Firstly, the input testing video sequences are converted to absolute difference images through Formula (2). Secondly, the absolute difference images are clipped to image patches. Different from video sequence preprocessing for training data set, data augmentation isn't applied to testing data set. We only draw three image patches from each of the absolute difference images. In other words, three image patches are clipped from each of the pristine frame and the tampered frame by the means described in Figure 1a.

After preprocessing, the testing image patches are input to the trained CNN-based model, and the classification results for each image patch are obtained. As described above, there are three image patches in each of video frames. If any one of the three image patches is predicted as tampered image patch, the video frames, which contained this image patch, should be marked as tampered frame. On the other hand, when all of the three image patches in a video frame are classified as pristine ones, this video frame should be labeled as pristine. Based on this classification rule, we can make a rough decision for each of the video frames.

We also deploy a very simple post-processing procedure for each of video frames to get a more accurate classification. This post-processing procedure uses a non-overlapping slide window to refine the previous rough decision. Let L denote the sliding window size, T represent the number of video frames labeled as tampered in the sliding window. Therefore, $L - T$ is the number of pristine video

frames in the same sliding window. In this paper, we set $L = 10$, and stride size of the sliding window to L, namely let all slide windows non-overlapping. In the post-processing procedure for more accurate classification, if $T \geq 7$, all of the video frames labeled as pristine in the sliding window are changed to tampered ones. On the contrary, if $T \leq 3$, all of the video frames labeled as tampered are changed to pristine ones.

3.3. Experimental Results

We compare our deep learning forgery detection approach with Chen et al.'s method in [3]. The following criteria defined in [3] are used in the experiments.

$$PFACC = \sum correctly\_classified\_pristine\_frames / \sum pristine\_frames$$
$$FFACC = \sum correctly\_classified\_forged\_frames / \sum forged\_frames$$
$$FACC = \sum correctly\_classified\_frames / \sum all\_the\_frames$$

where PFACC is *Pristine Frame Accuracy*, FFACC is *Forged Frame Accuracy* and FACC is *Frame Accuracy*. All of these are performance metrics for frame-type identification. After we use the non-overlapping slide window to get a more accurate classification, some frames in the forged video sequence, which were classified to incorrect labels, may be reassigned new labels. *Precision*, *Recall* and *F1 score* are used to evaluate the final classified accuracy for forged frames.

The performance comparison of experimental results are shown in Figure 3. We have repeated our experiment 10 times and obtained a total of 10 testing results. The testing results were computed as the mean and standard deviation of detection accuracy. As shown in Figure 3, the standard deviations of the average accuracy show the robustness of our approach. In order to make the lines in the figure easy to be discriminate, VACC *(Video Accuracy)* is not shown in Figure 3. The performance of VACC, which was greater than 99% achieved by Chen et al.'s method, is 100% by the proposed method.

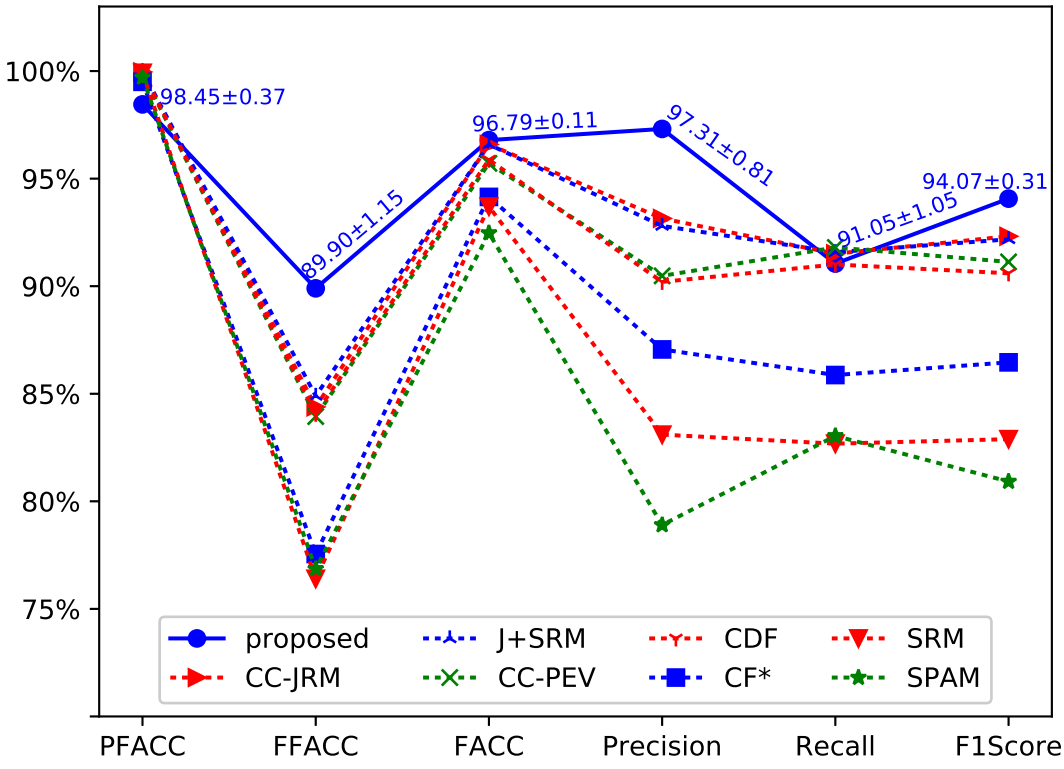

Figure 3. Detection performance of the proposed method compared with Chen et al.'s methods [3].

Chen et al. adopt seven steganalysis feature sets to achieve different performance. The seven steganalysis feature sets consist of CC-JRM [25], J+SRM [25], CC-PEV [26,27], CDF [28], CF* [29], SRM [21] and SPAM [30]. There are various performances achieved with using these feature sets. The accuracy rate provided by the seven algorithms can be found at Table II in [3]. As shown in Figure 3, the proposed method has achieved better performance than that achieved by using the steganalysis feature sets based approach in [3].

The proposed method is a deep learning approach for detection of object forgery in advanced video. Different from the traditional non-deep learning methods, the proposed method has the capability to automatically extract high-dimension features from the input image patches and make efficient representations. The traditional methods, such as the Chen et al.'s methods [3], could only use one type of artificial features to realize classification. As a result, we can see that our method is superior to the traditional methods.

4. Conclusions

In this paper, we developed an object forgery detection approach based on the deep convolutional neural network. The experimental results have shown that the proposed method achieves better performance than that reported in [3] on SYSU-OBJFORG, which is reported as the largest object forged video data set with advanced video encode frameworks. In the future, we will focus on the localization for forged region in each of the tampered video frames. Also, how to apply the trained CNN-based model to detect object forgery for lower bitrate video sequence or lower resolution video sequence, which is named as transfer learning [31–34] in deep learning research, would be another important future work.

Acknowledgments: This work was supported in part by the Public Technology Application Research Project of ZheJiang Province under Grant 2017C33146, in part by the Humanities and Social Sciences Foundation of Ministry of Education of China under Grant 17YJC870021, in part by the National Natural Science Foundation of China under Grant 61571139.

Author Contributions: Ye Yao conceived the overall idea of the article and wrote the paper. Yunqing Shi was the research advisor and provided methodology suggestions. Shaowei Weng and Bo Guan performed and carried out the experiments, analyzed the data, and contributed to the revisions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rocha, A.; Scheirer, W.; Boult, T.; Goldenstein, S. Vision of the unseen: Current trends and challenges in digital image and video forensics. *ACM Comput. Surv.* **2011**, *43*, 26–40.
2. Stamm, M.C.; Wu, M.; Liu, K.R. Information forensics: An overview of the first decade. *IEEE Access* **2013**, *1*, 167–200.
3. Chen, S.; Tan, S.; Li, B.; Huang, J. Automatic Detection of Object-Based Forgery in Advanced Video. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 2138–2151.
4. Tan, S.; Chen, S.; Li, B. GOP based automatic detection of object-based forgery in advanced video. In Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Hong Kong, China, 16–19 December 2015; pp. 719–722.
5. Qureshi, M.A.; Deriche, M. A bibliography of pixel-based blind image forgery detection techniques. *Signal Process. Image Commun.* **2015**, *39*, 46–74.
6. Birajdar, G.K.; Mankar, V.H. Digital image forgery detection using passive techniques: A survey. *Digit. Investig.* **2013**, *10*, 226–245.
7. Al-Qershi, O.M.; Khoo, B.E. Passive detection of copy-move forgery in digital images: State-of-the-art. *Forensic Sci. Int.* **2013**, *231*, 284–295.
8. Sitara, K.; Mehtre, B.M. Digital video tampering detection: An overview of passive techniques. *Digit. Investig.* **2016**, *18*, 8–22.

9. Bestagini, P.; Milani, S.; Tagliasacchi, M.; Tubaro, S. Local tampering detection in video sequences. In Proceedings of the IEEE 15th International Workshop on Multimedia Signal Processing (MMSP), Pula, Italy, 30 September–2 October 2013; pp. 488–493.
10. Lin, C.S.; Tsay, J.J. A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis. *Digit. Investig.* **2014**, *11*, 120–140.
11. Chen, R.; Yang, G.; Zhu, N. Detection of object-based manipulation by the statistical features of object contour. *Forensic Sci. Int.* **2014**, *236*, 164–169.
12. Pandey, R.C.; Singh, S.K.; Shukla, K.K. Passive copy-move forgery detection in videos. In Proceedings of the International Conference on Computer and Communication Technology (ICCCT), Allahabad, India, 26–28 September 2014; pp. 301–306.
13. Bondi, L.; Baroffio, L.; Güera, D.; Bestagini, P.; Delp, E.J.; Tubaro, S. First Steps Toward Camera Model Identification with Convolutional Neural Networks. *IEEE Signal Process. Lett.* **2017**, *24*, 259–263.
14. Tuama, A.; Comby, F.; Chaumont, M. Camera model identification with the use of deep convolutional neural networks. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, UAE, 4–7 December 2016; pp. 1–6.
15. Xu, G.; Wu, H.Z.; Shi, Y.Q. Structural Design of Convolutional Neural Networks for Steganalysis. *IEEE Signal Process. Lett.* **2016**, *23*, 708–712.
16. Yang, P.; Ni, R.; Zhao, Y. Recapture Image Forensics Based on Laplacian Convolutional Neural Networks. In Proceedings of the 15th International Workshop on Digital Forensics and Watermarking (IWDW), Beijing, China, 17–19 September 2016; pp. 119–128.
17. Bayar, B.; Stamm, M.C. A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer. In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security (IH&MMSec), Vigo, Spain, 20–22 June 2016; pp. 5–10.
18. Rao, Y.; Ni, J. A deep learning approach to detection of splicing and copy-move forgeries in images. In Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, UAE, 4–7 December 2016; pp. 1–6.
19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, Nevada, 3–6 December 2012; pp. 1097–1105.
20. Qian, Y.; Dong, J.; Wang, W.; Tan, T. Deep learning for steganalysis via convolutional neural networks. In Proceedings of the SPIE 9409, Media Watermarking, Security, and Forensics, San Francisco, CA, USA, 9–11 February 2015; Volume 9409, p. 94090J.
21. Fridrich, J.; Kodovsky, J. Rich Models for Steganalysis of Digital Images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882.
22. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 448–456.
23. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML), Haifa, Israel, 21–24 June 2010; pp. 807–814.
24. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. In Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, FL, USA, 3–7 November 2014; pp. 675–678.
25. Kodovsky, J.; Fridrich, J. Steganalysis of JPEG images using rich models. In Proceedings of the SPIE 8303, Media Watermarking, Security, and Forensics, Burlingame, CA, USA, 23–25 January 2012; p. 83030A.
26. Pevny, T.; Fridrich, J. Merging Markov and DCT features for multi-class JPEG steganalysis. In Proceedings of the SPIE, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, USA, 28 January 2007; Volume 6505, p. 650503.
27. Kodovskỳ, J.; Fridrich, J. Calibration revisited. In Proceedings of the 11th ACM Workshop on Multimedia and Security (MMSec), Princeton, NJ, USA, 7–8 September 2009; pp. 63–74.
28. Kodovsky, J.; Pevny, T.; Fridrich, J. Modern steganalysis can detect YASS. In Proceedings of the SPIE 7541, Media Forensics and Security II, San Jose, CA, USA, 18–20 January 2010; p. 754102.
29. Kodovsky, J.; Fridrich, J.; Holub, V. Ensemble classifiers for steganalysis of digital media. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 432–444.

30. Pevny, T.; Bas, P.; Fridrich, J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 215–224.
31. Shahriari, A. Distributed Deep Transfer Learning by Basic Probability Assignment. *arXiv*, **2017**, arXiv:1710.07437.
32. Nanni, L.; Ghidoni, S.; Brahnam, S. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognit.* **2017**, *71*, 158–172.
33. Galanti, T.; Wolf, L.; Hazan, T. A theoretical framework for deep transfer learning. *Inf. Inference J. IMA* **2016**, *5*, 159–209.
34. Long, M.; Zhu, H.; Wang, J.; Jordan, M.I. Deep Transfer Learning with Joint Adaptation Networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2208–2217.