symmetry

MDPI

Article

Frame Identification of Object-Based Video Tampering Using Symmetrically Overlapped Motion Residual

Tae Hyung Kim, Cheol Woo Park and Il Kyu Eom *

Department of Electronics Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan 46241, Korea; mbcthkim@pusan.ac.kr (T.H.K.); cwoop92@pusan.ac.kr (C.W.P.)
* Correspondence: ikeom@pusan.ac.kr; Tel.: +82-51-510-2414

Abstract: Image and video manipulation has been actively used in recent years with the development of multimedia editing technologies. However, object-based video tampering, which adds or removes objects within a video frame, is posing challenges because it is difficult to verify the authenticity of videos. In this paper, we present a novel object-based frame identification network. The proposed method uses symmetrically overlapped motion residuals to enhance the discernment of video frames. Since the proposed motion residual features are generated on the basis of overlapped temporal windows, temporal variations in the video sequence can be exploited in the deep neural network. In addition, this paper introduces an asymmetric network structure for training and testing a single basic convolutional neural network. In the training process, two networks with an identical structure are used, each of which has a different input pair. In the testing step, two types of testing methods corresponding to two- and three-class frame identifications are proposed. We compare the identification accuracy of the proposed method with that of the existing methods. The experimental results demonstrate that the proposed method generates reasonable identification results for both two- and three-class forged frame identifications.

Keywords: object-based video tampering; frame identification; motion residual; convolutional neural network; symmetrically overlapped motion residual; three-class identification

Citation: Kim, T.H.; Park, C.W.; Eom, I.K. Frame Identification of Object-Based Video Tampering Using Symmetrically Overlapped Motion Residual. *Symmetry* **2022**, *14*, 364. https://doi.org/10.3390/sym14020364

Academic Editor: Kuo-Hui Yeh

Received: 18 January 2022
Accepted: 10 February 2022
Published: 12 February 2022

1. Introduction

Millions of images and videos are created and distributed through the Internet every day. A wide range of editing tools is available even to non-professionals so that they can easily edit videos on their computers and mobile phones. Therefore, videos are often used after undergoing an editing process rather than being used as is. Because most of the editing is of good quality, there may be times when the reliability of the video is questionable. When videos used as evidence in courts or as news to report facts by the media are tampered with for malicious purposes, the consequences often cause serious social problems or harm. For this reason, video forgery detection has emerged as an important topic [1,2].

Earlier video tampering methods were limited to the temporal domain, referred to as frame-based forgery, which included frame insertion, frame deletion, frame duplication, and double-compression. In the frame insertion forgery, a set of frames is inserted in the order of original frames. When one or more frames are deleted from a sequence of frames in a video, it is known as frame deletion. The size of the frame-deleted video is generally less than that of the original video. Frame duplication means copying a set of frames in a video and pasting that into the same video. Over the years, several techniques to detect frame-based video forgery were proposed.

Earlier, double-compression detection algorithms [3–6] were presented. However, to tamper with a video sequence, the video should first be decompressed. After the manipulation, the video is recompressed, irrespective of whether the frame has been tampered with. Therefore, double-compression detection is performed in combination with frame insertion

and deletion in frame-based forgery detection methods. Stamm et al. [7] proposed a temporal anti-forensic algorithm for motion compensated videos. They developed a theoretical model of forensically detectable fingerprints and used the model to detect video frame deletion or addition. Gironi et al. [8] introduced a frame insertion and deletion algorithm of whole frames in digital videos using double encoding detection. This method can be applied when different codecs are used for the first and second compressions. Yang et al. [9] presented a two-step approach for detecting frame duplication using similarity analysis. In the first step, the features of each frame are extracted via singular value decomposition. In addition, the Euclidean distance is calculated between features of each frame and the reference frame to select candidate duplication frames. In the second step, the candidate duplications are confirmed through random block matching.

Object-based video forgery is a common tampering method by adding new objects to a video sequence or removing existing ones. However, object-based video forgery and detection have only recently attracted attention. Because this tampering method is performed in the spatial domain of the video, it can be considered as copy-move or splicing forgery of images. However, applying the image-based forgery detection techniques [10–12] to each frame of the video is not appropriate because they cannot exploit the temporal information in compressed videos.

In 2014, Richao et al. [13] proposed an object-based video manipulation detection algorithm using statistical features of an object contour. The algorithm extracted several statistical features, such as the moment features of detailed wavelet coefficients and the average gradient of each color channel, and inserted them into a support vector machine (SVM) as feature vectors for classifying natural and forged objects. Tan et al. [14] presented a GOP-based automatic detection method for object-based video forgery. This technique extracted steganalysis features [15] from motion residuals and employed an ensemble classifier [16] to construct a frame manipulation detector. Similar to the method in [14], Chen et al. [17] extracted 548-dimensional image steganalysis features [18] from noise patterns based on motion residuals of frame images and trained the features to identify object-based forged frames using the ensemble classifier. Saddique et al. [19] exploited the chrominance value of consecutive frame difference and discriminative robust local binary pattern to model the inconsistency embedded in the video frames due to forgery and used SVM to detect whether the pair of consecutive frames was forged. Sitara et al. [20] proposed a video tampering detection method using differentiating optical camera zooming from synthetic zooming based on pixel variance correlation and sensor pattern noise. Aloraini et al. [21] introduced a new video tampering detection and localization approach based on sequential and patch analyses performed by modeling video sequences as stochastic processes and as a mixture model of normal and anomalous patches. Experimental results showed that this approach achieves excellent detection performance with low computational complexity.

Recently, deep learning has been in the spotlight in the field of computer vision [22]. Deep learning-based methods such as CNN have attained great success in the domain of image processing and computer vision. The reason is that deep neural networks are capable of extracting problem-specific and complex high-dimensional features to efficiently represent the information needed. In the field of video forensics, several studies focusing on the convolutional neural network (CNN) have been conducted. Yao et al. [23] extracted high-dimensional features based on high-frequency signals from motion residual features and used CNN to determine whether the frames were tampered with. Kohli et al. [24] proposed a spatiotemporal method using CNN to detect video tampering and to localize the forged region in a forged frame. They employed the motion residual to train the presented network. Yang et al. [25] proposed a spatiotemporal trident network based on the spatial rich model and three-dimensional convolution, which provides branches and can theoretically improve the detection and localization accuracy of tampered regions.

Motion residual was widely used as a key feature to detect object-based video forgery [17,23–25]. For each video frame, the motion residual contains a substantial number of the intra-frame properties of that frame as well as the inter-frame inherent properties of

the corresponding frame. Because the motion residual contains both the intra- and inter-frame inherent properties of the corresponding frames, it is frequently used for detecting the object-based video forgery. However, it is often difficult to distinguish a motion induced by tampering from the one caused by a scene change. The conventional motion residual features cannot sufficiently distinguish between these two differences.

Most past studies [21,23,25] performed two-class identification, which classifies whether the video frame has been tampered with, to detect object-based video forgery. In [24], another two-class identification is defined by identifying whether the frame has been forged or double-compressed. Only the method of Chen et al. [17] reported the three-class identification results.

In this paper, we propose a novel video frame identification algorithm for object-based video tampering detection. The main contributions of this paper are as follows.

(1) A symmetrically overlapped motion residual is proposed to enhance the discernment of video frames. Three different motion residuals are presented on the basis of overlapping temporal frames. The proposed overlapped motion residuals can reduce natural motions while preserving motions caused by tampering. Because the overlapped motion residuals use three different temporal windows, temporal changes in motion residuals can be exploited in the neural network.
(2) An asymmetric CNN structure for training and testing is proposed. In the training step, we design a single basic CNN and use it to perform two different training processes, resulting in two trained networks with an identical structure. In the testing process, we propose two types of testing methods corresponding to two- and three-class frame identifications. Two-class identification determines whether a video frame has been forged using a single trained network. In contrast, three-class identification is also performed to classify whether a frame is pristine, double-compressed, or tampered with by combining the two learned basic networks.

The proposed method generates reasonable identification results for both two- and three-class forged video frame identifications.

The remainder of this paper is organized as follows. Section 2 introduces the basic process of the object-based video forgery and detection. Section 3 presents the proposed symmetrically overlapped motion residuals. The proposed frame identification network structure is discussed in Section 4. In Section 5, the performance of the proposed method is compared with that of existing methods using the experimental results. Section 6 presents the discussion, and the conclusions are presented in Section 7.

2. Object-Based Video Forgery and Detection

Object-based video forgery refers to adding new objects to or removing existing objects from a video frame. Figure 1 illustrates an example of the object-based video tampering process on a sample video sequence. Figure 1a shows four consecutive frames of a pristine video, and Figure 1b contains two tampered-with frames. The first two frames in Figure 1b are not tampered with, and objects are removed from the last two frames. The objects surrounded by the red ellipse in the first and second frames remain unchanged, and the objects in the third and fourth frames have disappeared. Figure 1b presents a typical example of video tampering by object removal.

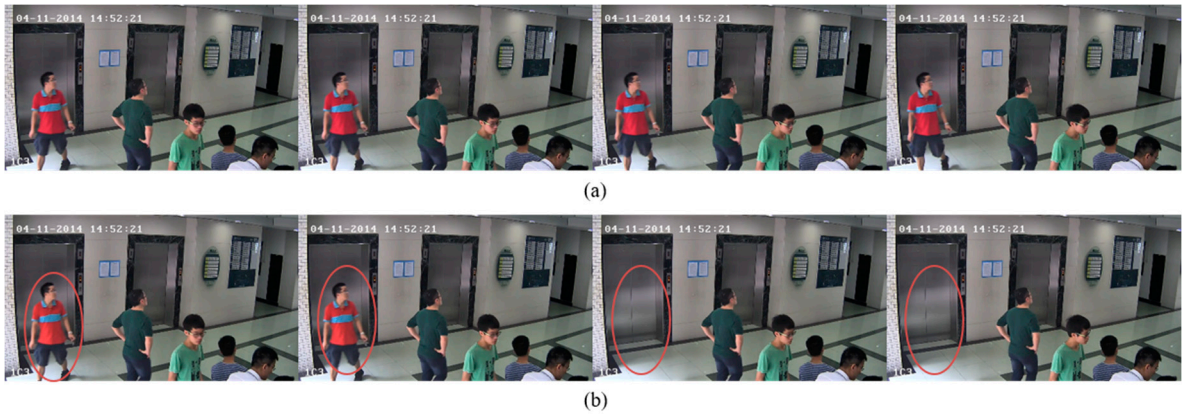


Figure 1. Example of object-based video tampering. (**a**) Pristine frames. (**b**) Partially tampered-with frames. The object surrounded by the red ellipse is removed from the third and fourth frames.

Because all videos are compressed for storage and transmission, they should first be decompressed in order to tamper with or just enjoy them. When there is an intention to tamper with the video sequence, forgery is performed on specific frames of the recompressed video, and there are frames that have not been manipulated. At the end of the tampering process, the entire video sequence is recompressed. Videos intended to be manipulated are first decompressed, and only specific frames are tampered with in most cases. After this process, the tampered with and non-tampered with frames are recompressed simultaneously. Video frames that are not tampered with undergo compression twice, and manipulated frames are compressed again after forgery. In this scenario, there are three types of frames: (1) pristine frame, (2) double-compressed frame without tampering, and (3) tampered frame. On the basis of this scenario, there are two types of frame identification, such as two-class identification (forged and non-forged) and three-class identification (pristine, double-compressed, and forged). Figure 2 shows two types of frame identification to detect object-based video forgery. This paper presents both two- and three-class frame identification results using various combinations of a single CNN structure.

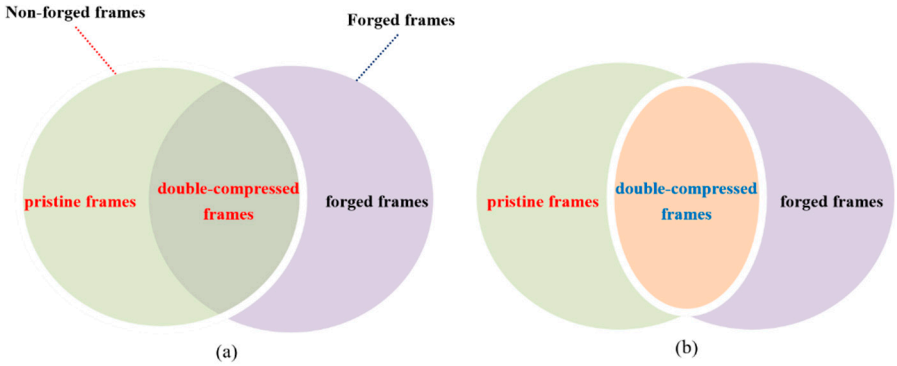


Figure 2. Two types of frame identification in object-based video tampering. (**a**) Two-class identification. (**b**) Three-class identification.

3. Symmetrically Overlapped Motion Residual

A video is defined in both spatial and temporal domains. In the spatial domain, the intra-frame properties are almost identical to those of a conventional image. However, since motion compensation is applied to the video coding, temporal information is added to the intra-frame. In contrast, the inter-frame properties describe the temporal characteristics. The strong correlation between adjacent frames in a video implies that each frame in a local temporal window comprises two parts: the motion part and the static part [17]. The static part is identical to the base frame of a local temporal window, whereas the

motion part is the motion residual for that base frame. For each frame, the motion residual contains many of the intra-frame properties as well as the inter-frame properties of the corresponding frame because it represents the temporal changes from that frame to the base frame. Because the motion residual contains both intra- and inter-frame inherent properties of the corresponding frame, it is used as a primal feature for object analysis. Therefore, motion residual is a commonly used feature to detect video forgery in several studies [17,21,23,24]. In this section, we briefly review the motion residual and propose a symmetrically overlapped motion residual to improve the frame identification performance.

3.1. Motion Residual

Let V be the decompressed video sequence defined as

$$V = \left\{ \mathbf{f}^{(1)}, \cdots, \mathbf{f}^{(Q)} \right\}, \tag{1}$$

where $\mathbf{f}^{(k)}$ ($k = 1, \ldots, Q$) is the k-th decompressed video frame and Q is the number of frames of the video sequence. In a temporal window, a collusion operation can be defined around a center frame $\mathbf{f}^{(k)}$ with the temporal window size of $L = 2l + 1$, where l is the number of the left and right neighbors of $\mathbf{f}^{(k)}$. Figure 3 shows the frame structure for collusion operation, which is performed using l frames to the left and right around the base frame $\mathbf{f}^{(k)}$. Let $\mathbf{c}^{(k)}$ be the colluded result for $\mathbf{f}^{(k)}$, which is defined as

$$\mathbf{c}^{(k)} = \mathrm{C}\left[\mathbf{f}^{(k-l)}, \cdots, \mathbf{f}^{(k)}, \cdots, \mathbf{f}^{(k+l)} \right], \tag{2}$$

where $\boldsymbol{C}$ is the collusion operator.

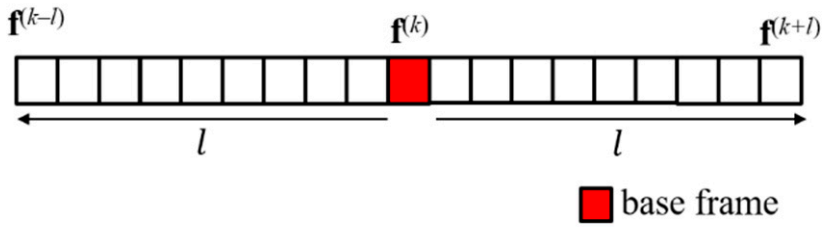


Figure 3. Frame structure for collusion operation.

In the video forgery detection approaches, the minimum or the median operation is commonly used as the collusion operator. Let $f^{(k)}(i,j)$ and $c^{(k)}(i,j)$ be the pixel values of $\mathbf{f}^{(k)}$ and $\mathbf{c}^{(k)}$ at the spatial location (i,j), respectively. The colluded result at the location (i,j) is

$$c^{(k)}(i,j) = \boldsymbol{C}\left[f^{(k-l)}(i,j), \cdots, f^{(k)}(i,j), \cdots, f^{(k+l)}(i,j) \right]. \tag{3}$$

Figure 4 shows an example of obtaining $\mathbf{c}^{(k)}$ using median collusion operation.

The motion residual image is defined as the absolute difference between $\mathbf{f}^{(k)}$ and $\mathbf{c}^{(k)}$ as follows.

$$\mathbf{r}^{(k)} = \left| \mathbf{f}^{(k)} - \mathbf{c}^{(k)} \right|, \tag{4}$$

where $\mathbf{r}^{(k)}$ is the motion residual of the k-th frame and $|\cdot|$ denotes the absolute value. The motion residual is considered a measure of the motion or temporal change in the temporal window. Adding or removing objects from a certain frame can cause an abrupt change in a video sequence. Because the motion residual can act as a clue in the detection of the object-based frame tampering, it is frequently used in detecting video tampering [17,23–25]. However, a scene change in a video frame can also cause an abrupt change. Thus, it is important to distinguish between the motion residual by a scene change and that by tampering.

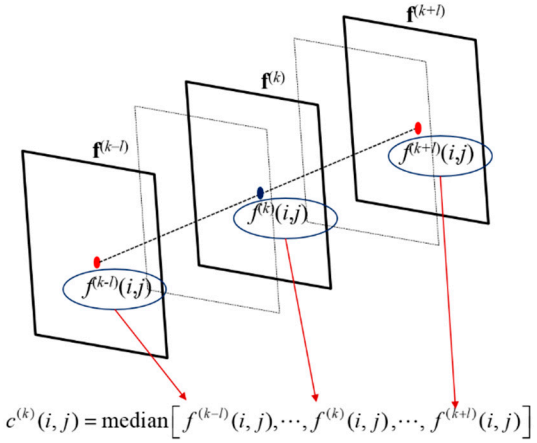


Figure 4. Example of the median collusion operation at pixel level.

3.2. Proposed Overlapped Motion Resiual

It is important to increase the difference between natural motion and the motion created by object adding or removing forgery. This paper presents an overlapped motion residual to improve the discernment of the motion residual. The proposed overlapped motion residual has three motion residual frames, including base, left, and right motion residuals with a temporal window size of $L_p = 2l_p + 1$, as illustrated in Figure 5. In conventional motion residual, l is set to 9, therefore, the size of the temporal window is $L = 2l + 1 = 2 \times 9 + 1 = 19$. In our method, l_p is set to 5, which results in the temporal window size of $L_p = 11$.

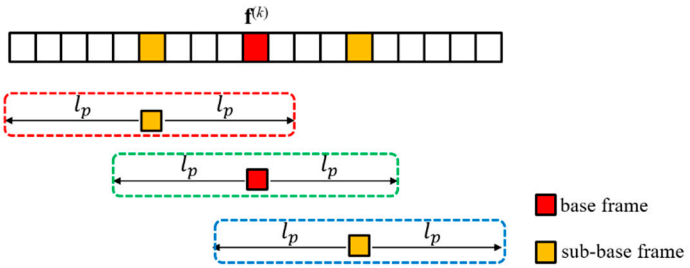


Figure 5. Frame structure for overlapped motion residual generation.

The three colluded results, $\mathbf{c}_{base}{}^{(k)}$, $\mathbf{c}_{right}{}^{(k)}$, and $\mathbf{c}_{left}{}^{(k)}$ for $\mathbf{f}^{(k)}$ are defined as

$$\mathbf{c}_{base}{}^{(k)} = \boldsymbol{C}\left[\mathbf{f}^{(k-l_p)},\cdots,\mathbf{f}^{(k)},\cdots,\mathbf{f}^{(k+l_p)}\right], \tag{5}$$

$$\mathbf{c}_{right}{}^{(k)} = \boldsymbol{C}\left[\mathbf{f}^{(k-1)},\cdots,\mathbf{f}^{(k+l_p-1)},\cdots,\mathbf{f}^{(k+2l_p-1)}\right], \tag{6}$$

$$\mathbf{c}_{left}{}^{(k)} = \boldsymbol{C}\left[\mathbf{f}^{(k-2l_p+1)},\cdots,\mathbf{f}^{(k-l_p+1)},\cdots,\mathbf{f}^{(k+1)}\right]. \tag{7}$$

To obtain the proposed overlapped motion residuals, we use the median filter for the collusion operation. In conclusion, we have three motion residuals $\mathbf{r}_{base}{}^{(k)}$, $\mathbf{r}_{right}{}^{(k)}$, and $\mathbf{r}_{left}{}^{(k)}$ corresponding to $\mathbf{c}_{base}{}^{(k)}$, $\mathbf{c}_{right}{}^{(k)}$, and $\mathbf{c}_{left}{}^{(k)}$, respectively as follows.

$$\mathbf{r}_{base}{}^{(k)} = \left|\mathbf{f}^{(k)} - \mathbf{c}_{base}{}^{(k)}\right|, \tag{8}$$

$$\mathbf{r}_{right}{}^{(k)} = \left|\mathbf{f}^{(k+l_h-1)} - \mathbf{c}_{right}{}^{(k)}\right|, \tag{9}$$

$$\mathbf{r}_{left}{}^{(k)} = \left|\mathbf{f}^{(k-l_h+1)} - \mathbf{c}_{left}{}^{(k)}\right|. \tag{10}$$

All motion residuals include base frame $\mathbf{f}^{(k)}$, and the base motion residual, $\mathbf{r}_{base}{}^{(k)}$ contains two sub-base frames. This why the proposed method is called overlapped motion residual.

Figure 6a presents the conventional and overlapped motion residual images for a sample pristine frame. As shown in Figure 6a, there is no significant difference between conventional motion residual and the overlapped motion residuals. Because the proposed overlapped motion residual uses a smaller temporal window than the conventional motion residual, fine background noises are shown in the overlapped motion residuals. Figure 6b presents the forged frame by removing two objects from the pristine frame shown in Figure 6a. The forged trace is not significantly reinforced in the conventional motion residual image. However, the trace of object tampering appears relatively prominent in the overlapped motion residuals. Because the motion in this example exists from left to right, the tampering trace is clearly visible, especially at the left overlapped motion residual $\mathbf{r}_{left}{}^{(k)}$. The fine details of background due to the small temporal window can cause a relatively strengthening effect into the tampered-with area. Because the proposed method uses three different temporal windows, the temporal change of the motion residual can be more effectively reflected in the deep neural network.

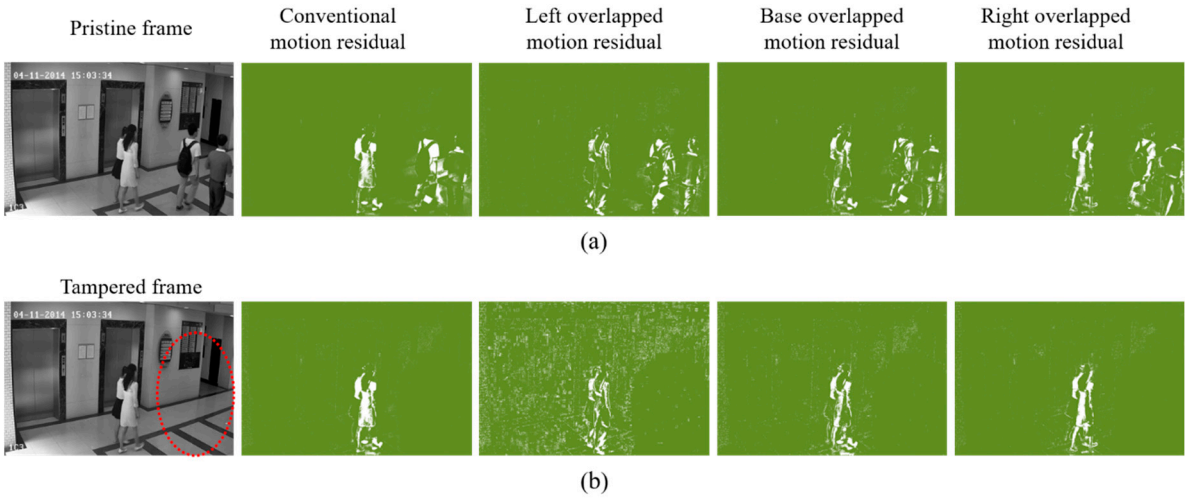


Figure 6. Sample motion residual images for pristine and corresponding tampered-with frame. (**a**) Conventional motion residual and overlapped three motion residual images for pristine frame; (**b**) Conventional motion residual and overlapped three motion residual images for tampered-with frame. All motion residual images are rescaled to a maximum value of 255.

3.3. Patch Generation of Overlapped Motion Resiual

Because the video is partially tampered with, the number of pristine frames is usually greater than that of forged frames. Therefore, there exists an imbalance between data with different characteristics. In addition, a data augmentation technique to increase artificially the amount of data to reduce network overfitting is needed. Yao et al. [23] introduced an asymmetric data augmentation method to increase the amount of data in the training process and resolved the data imbalance. In this study, we adopt the asymmetric data augmentation method of Yao et al. to increase the amount of data and resolve imbalance.

Let the size of a single frame be $M \times N$. The size of overlapped motion residual of one video frame is also $M \times N$. The motion residual image is first cropped to an $m \times m$ patch at the left-most position, where m is calculated as

$$m = \min(M, N). \tag{11}$$

Next, new patches are created by striding to the right until a predefined number of patches are satisfied. Using the asymmetric data augmentation technique, the stride size t is calculated for a given patch size p as follows.

$$t = \left\lfloor \frac{\max(M, N) - \min(M, N)}{p} \right\rfloor, \tag{12}$$

where $\lfloor \rfloor$ is the floor function. Different numbers of patches for the three classes to be distinguished are used in this paper.

For an $m \times m$ patch, maximum pooling is used to reduce excessive computation time as well as to make the network robust to variations in overlapped motion residual values. The patch size is reduced to $(m/w) \times (m/w)$ with a maximum pooling window size of $w \times w$. In this study, the size of w is set to 3. A high-frequency filter is frequently exploited to enhance a weak feature for various machine-learning applications [23,26,27]. The motion residual patch that has undergone the maximum pooling process is passed through the high-frequency filter with a 5×5 shift-invariant convolution kernel [26] before feeding to the proposed network.

4. Proposed Frame Identification

In this paper, we propose an asymmetric network structure for training and testing. In the training process, two training pairs, such as forged and non-forged patches, and pristine and double-compressed patches, are fed to the same basic CNN model. The training pair with the forged and non-forged patches is used for two-class identification in the testing process, as shown in Figure 2a. The class of non-forged patch is divided into pristine and double-compressed patches. Using this relationship, we can identify a three-class problem as forged, double-compressed, and pristine in the testing process, as shown in Figure 2b.

4.1. Basic Network

Figure 7 shows a schematic of the basic CNN architecture, which is a simple and typical CNN model in which a similar convolution layer is repeated five times. The input of this network is three-dimensional overlapped motion residual patches after maximum pooling and high-frequency filtering. Every convolution layer includes a batch normalization (BN), rectified linear unit (ReLU), and average pooling (AP).

The kernel sizes of the five convolution layers are 3×3, 3×3, 3×3, 1×1, and 1×1. The number of kernels for convolution layers I, II, II, IV, and V is 8, 16, 32, 64, and 128, respectively. The kernel and stride sizes of AP I are 5×5 and 2, respectively. AP II has a kernel size of 15×15 and stride size of 1. Detailed information of the basic network is summarized in Table 1. The overlapped motion residual patch has a size of $3 \times (m/w) \times (m/w)$. In Table 1, let $m' = m/w$. Because the stride size of AP I is 2, the patch is reduced to the size of $1/2^2$. In AP II, the global average pooling with kernel size of $(m' \times m')/2^{10}$ results in a 128 dimensional final feature vector inserted to the fully connected network. In this paper, we use 1280×720 size video sequence and 3×3 window for the maximum pooling. Therefore, $m' = 720/3 = 240$.

Table 1. Detailed information of the basic network.

	Kernel Size	Number of Kernel	Stride Size	Feature Size after AP I
Layer I	3×3	8	1	$8 \times (m' \times m')/22$
Layer II	3×3	16	1	$16 \times (m' \times m')/24$
Layer III	3×3	32	1	$32 \times (m' \times m')/26$
Layer IV	1×1	64	1	$64 \times (m' \times m')/28$
Layer V	1×1	128	1	$128 \times (m' \times m')/210$
AP I	5×5	1	2	-
AP II	$(m' \times m')/210$	1	Global	128

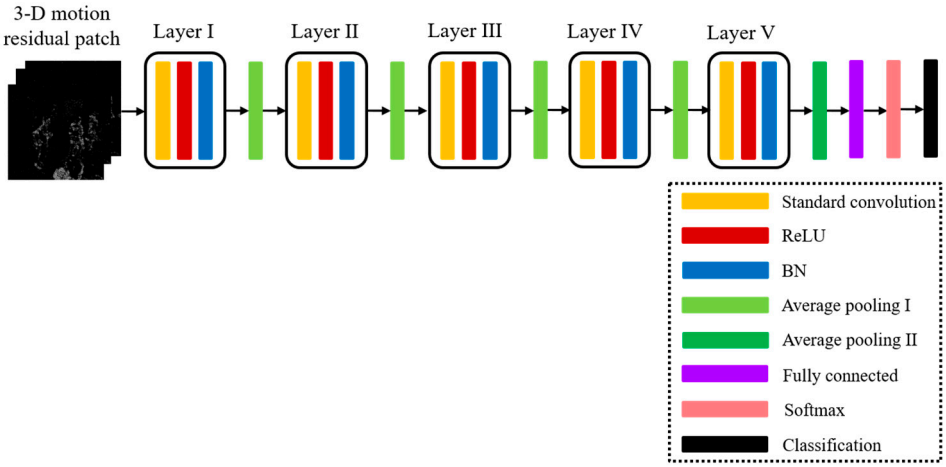


Figure 7. Basic convolutional neural network (CNN) structure.

4.2. Training

A video frame can roughly be divided into two types: forged and non-forged. In this paper, we further divided the non-forged frame into pristine and double-compressed. In the training process, two machine learning networks are configured. The first network is trained with forged and non-forged patches as shown in Figure 8a. The number of patches in one forged frame is set to 9 and the number of patches in a single non-forged frame is set to 3. Because the number of patches in the forged frame is generally smaller than that in the non-forged frame, the number of patches in the forged frame is set to be greater than that in the non-forged frame. Figure 8b presents the basic CNN for distinguishing pristine and double-compressed frames. The number of patches in a single frame for pristine and double-compressed frames is set to 3 and 5, respectively.

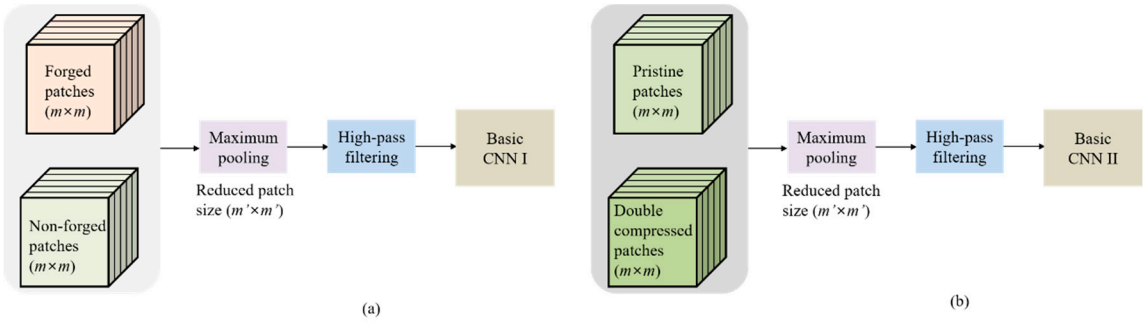


Figure 8. Configuration of machine learning network. (**a**) Basic network for classifying forged and non-forged patches; (**b**) Basic CNN for classifying pristine and double-compressed patches.

The loss function for the proposed deep learning model is the cross-entropy function for a 1-of-*G* coding scheme [28]. This loss function is

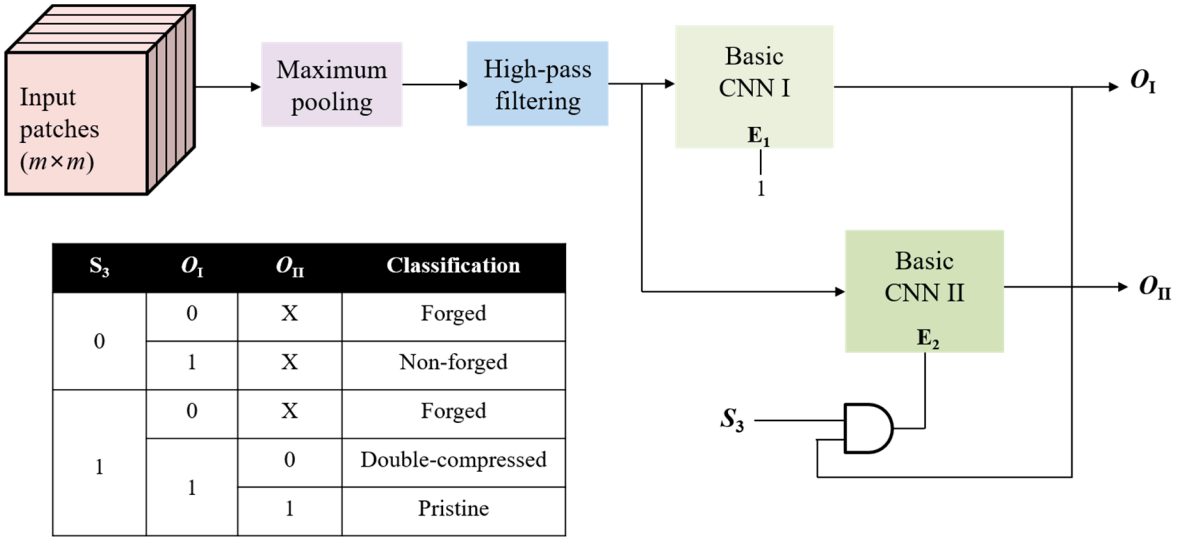
$$\text{Loss} = -\frac{1}{D}\sum_{d=1}^{D}\sum_{g=1}^{G}\omega_g \lambda_{dg} \ln y_{dg}, \tag{13}$$

where D is the number of input samples, G is the number of classes, ω_g is the weight for class g, λ_{dg} is the indicator that the d-th sample belongs to the g-th class, and y_{dg} is the output probability for sample d for class g.

4.3. Classification

The number of patches for a frame was set differently, depending on the type of frame in the training process. However, the type of frame cannot be known during the

testing process. Therefore, we fix the number of patches extracted from a single frame to 3. In the proposed method, both two-class identification (forged or not) and three-class identification (forged, double-compressed, or pristine) can be solved using the two learned basic networks. The identification process is performed in two stages. We try to explain the identification process using the concept of a logic circuit as shown in Figure 9.



S_3	O_I	O_{II}	Classification
0	0	X	Forged
	1	X	Non-forged
1	0	X	Forged
	1	0	Double-compressed
		1	Pristine

Figure 9. Classification method explained using a logic circuit concept.

In Figure 9, if S_3 is set to 0, the network performs two-class identification. In this situation, basic CNN II is disabled. According to the value of O_I, the input frame is classified as forged or non-forged. The three-class identification is performed when S_3 is set to 1. In this case, if $O_I = 0$, the network determines that the input frame has been forged, and basic CNN II is disabled. In contrast, if $O_I = 1$, basic CNN I determines that the input frame has not been tampered with, and basic CNN II is enabled. Then, we can classify the input frame as double-compressed if $O_{II} = 0$, and pristine if $O_{II} = 1$. Basic CNN I is always activated ($E_1 = 1$), and CNN II is enabled if both $S_3 = 1$ and $O_{II} = 1$ ($E_2 = S_3 \cdot O_{II}$).

In the testing process, a single frame is sliced into three patches. The tampered-with object usually occupies a small part of the frame. Thus, the network classifies the input frame as forged when only one of the three patches of a single frame has been tampered with. In basic CNN II, the majority vote for the three patches is used to determine whether the input frame is pristine or double-compressed.

5. Experimental Results

5.1. Dataset

For object-based video-tampering identification, there are three representative datasets, namely SYSU-OBJFORG [17], SULFA [29], and REWIND [30]. SULFA has only five videos with a resolution of 320 × 240. REWIND is based on the SULFA dataset with the same resolution as SULFA. The amount of data available for SULFA and REWIND is too small, and thus they are not sufficient for deep learning approaches in training and testing. Therefore, SULFA and REWIND are known to be insufficient in forgery detection due to the small number of samples and low image quality. SYSU-OBJFORG comprises 100 pristine video sequences and 100 tampered-with video sequences. Out of 100 tampered-with videos, 80 have one forged part, and the remaining 20 videos have two manipulated parts. Each video sequence has a duration of approximately 11 s, with a resolution of 1280 × 720 and a frame rate of 25 FPS. All videos are compressed in the H.264/MPEG-4 encoding format with a bitrate of 3 Mbps. In the experiment, 50 pairs of pristine and tampered-with video sequences are used for training, and the remaining 50 pairs of videos are used for testing.

In the training step, 50 pairs of videos are divided into training and validation data. One of the five parts is used for validation, and the remainder are used for training.

5.2. Experimental Setup

The proposed method is implemented using MATLAB R2019a and executed on a PC environment of Intel(R) Core(TM) i5-8500 3.00 GHz CPU and NVIDIA GeForce GTX 1050 GPU. The training options of the proposed deep learning model are stochastic gradient descent with momentum, initial learning rate of 0.001, learning rate drop factor of 0.2, learning rate drop period of 5, and batch size of 64 for both basic CNN I and CNN II. The training time of CNN I is 3794 min and CNN II takes 1488 min.

To evaluate the performance of the proposed method, we first define seven evaluation metrics: pristine frame accuracy (*PFACC*), forged frame accuracy (*FFACC*), double-compressed frame accuracy (*DFACC*), total frame accuracy (*TFACC*), *Precision*, *Recall*, and *F*1 *score*. These metrics are defined as

$$PFACC = \frac{\#\,\text{Correctly classifed pristine frames}}{\#\,\text{Pristine frames}}, \tag{14}$$

$$FFACC = \frac{\#\,\text{Correctly classifed forged frames}}{\#\,\text{Forged frames}}, \tag{15}$$

$$DFACC = \frac{\#\,\text{Correctly classifed double}-\text{compressed frames}}{\#\,\text{Double}-\text{compressed frames}}, \tag{16}$$

$$TFACC = \frac{\#\,\text{Correctly classifed frames}}{\#\,\text{All the frames}}, \tag{17}$$

$$Precision = \frac{TP}{TP + FP}, \tag{18}$$

$$Recall = \frac{TP}{TP + FN}, \tag{19}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{20}$$

where # represents the number of the set elements, *TP* is the number of true positives, *FP* is the number of false positives, and *FN* is the number of false negatives. In this study, imbalanced data are used in the testing process. The numbers of pristine frames, double-compressed frames, and forged frames are 14,217, 8420, and 5790, respectively. Therefore, we use averaged *Recall* and *Precision*.

5.3. Results

5.3.1. Two-Class Identification

Most video tampering detection methods classify video frames as forged and non-forged. Non-forged and just double-compressed frames are considered pristine frames. In the proposed network, two-class identification can be achieved with setting $S_3 = 0$ as shown in Figure 9. In this case, only basic CNN I is activated. Table 2 presents the confusion matrix for two-class identification obtained using the proposed network. From Table 2, we can see that both the non-forged and tampered-with frames are identified well with an accuracy of 98% or more.

Table 2. Confusion matrix for two-class forged frame identification based on the proposed method. The number in bracket indicates the number of frames used in the testing process.

Actual Frame / **Labeled Frame**	**Non-Forged**	**Forged**
Non-forged	98.10% (22,207)	1.90% (430)
Forged	1.93% (112)	98.07% (5678)

We compared our method with five state-of-the-art methods by Yao et al. [23] (CNN), Kohli et al. [24] (Temporal CNN), Aloraini et al. [31] (TF-SA), Aloraini et al. [21] (S-PA), and Yang et al. [25] (STN). Table 3 shows six metrics for six forgery detection algorithms, including the proposed method. CNN [23] exhibits good performance in terms of *PFACC;* however, it generates low *FFACC*. *FFACC* is the most important metric in identifying video frames because lower *FFACC* can cause serious problems. Therefore, the CNN method cannot be successfully applied to video-tampering identification. STN [25] achieves the best performance for all metrics as shown in Table 2. However, STM can only be applied to two-class problems. The proposed method ranks in the second place for various metrics, except *PFACC*.

Table 3. Six evaluation metrics for various methods. Numbers in bold indicate the highest performance and those in italics represent the second place.

Method	*PFACC*	*FFACC*	*TFACC*	*Precision*	*Recall*	*F1 Score*
CNN [23]	*98.45*	89.90	96.79	91.05	97.31	94.07
Temporal CNN [24]	-	96.04	97.49	-	-	-
TF-SA [31]	-	-	-	93.90	93.90	93.30
S-PA [21]	-	-	-	95.51	94.44	94.97
STN [25]	**99.50**	**98.75**	**99.34**	**98.14**	**98.75**	**98.44**
Proposed	98.10	*98.07*	*98.09*	96.23	*98.08*	*97.15*

5.3.2. Three-Class Identification

The three-class identification can be achieved by setting $S_3 = 1$ in the proposed approach as shown in Figure 9. Table 4 shows the confusion matrix for three-class identification obtained using the proposed deep network. As illustrated in Table 4, there is no misclassification of pristine as forged, and of forged as pristine. However, more than 5% of double-compressed is incorrectly classified as forged. The double-compressed frames seem to have a tendency to be identified as tampered-with frames because the discontinuity increases because of the accumulated compression errors.

Table 4. Confusion matrix for three-class video frame identification obtained using the proposed method. Numbers in bracket indicate the number of frames used in the testing process.

Actual Frame / Labeled Frame	Pristine	Double-Compressed	Forged
Pristine	99.89% (14,205)	0.11% (12)	0.00% (0)
Double-compressed	0.27% (23)	94.62% (7967)	5.11% (430)
Forged	0.00% (0)	1.93% (112)	98.07% (5678)

Only the method of Chen et al. [17] (ADOBF) provides both two- and three-class video forgery identification results. They use seven state-of-the-art steganalysis features, namely CC + PEV [18], SPAM [32], CDF [33], CF* [16], SRM [27], CC-JRM [34], and J + SRM [34], extracted from the conventional motion residuals. Table 5 shows seven evaluation metrics in three-class object-based video-tampering identification. As shown in Table 5, CC + JRM and J + SRM features achieve high *PFACC* and *DFACC* metrics while exhibiting poor *FFACC*. In contrast, the proposed method achieves low *DFACC* because 5.11% of double-compressed frames are misclassified as tampered-with frames and 0.27% of these frames are regarded as pristine frames. All *FFACC* values except ours exhibit low scores compared to the other metrics. However, the proposed method achieves the highest *FFACC*, *Precision*, *Recall*, and *F1 score*. Therefore, the proposed method can be efficiently applied to frame identification of object-based video forgery.

Table 5. Seven evaluation metrics for various methods in three-class identification. Numbers in bold indicate the highest performance and those in italics represent the second place.

Method	*PFACC*	*FFACC*	*DFACC*	*TFACC*	*Precision*	*Recall*	*F1 Score*
ADOBF [17]: CC + PEV [18]	99.90%	83.94%	95.22%	95.71%	90.48%	*91.80%*	91.13%
ADOBF [17]: SPAM [32]	99.71%	76.86%	89.03%	92.47%	78.90%	83.04%	80.92%
ADOBF [17]: CF* [33]	99.50%	77.55%	93.64%	94.15%	87.06%	85.87%	86.46%
ADOBF [17]: CDF [16]	*99.96%*	84.07%	95.67%	95.88%	90.20%	91.01%	90.60%
ADOBF [17]: SRM [27]	99.91%	76.40%	93.21%	93.70%	83.10%	82.86%	82.89%
ADOBF [17]: CC-FRM [34]	*99.96%*	*84.93%*	**97.82%**	96.59%	*93.15%*	91.51%	92.32%
ADOBF [17]: J + SRM [34]	**99.99%**	84.90%	*97.56%*	96.59%	92.80%	91.58%	92.18%
Proposed	99.89%	**98.07%**	94.62%	**97.97%**	**97.08%**	**97.53%**	**97.30%**

6. Discussion

In the field of video-frame tampering detection, three-class identification is more useful than two-class identification. However, most methods use only two-class identification except for the Chen et al. method [17]. Because the proposed approach uses both two- and three-class identifications, it can be applied to tampered-with video-frame classification. The fact that a frame is not tampered with but only double-compressed can be an indication that there is a possibility of manipulation of the video sequence to which the frame belongs. Therefore, detecting not only forged but also double-compressed frames can be useful evidence for video-tampering detection. From Tables 4 and 5, we can see that most of double-compressed frames are misclassified as tampered-with frames in the proposed approach. Determining what has not been forged as tampered with can cause some costs; however, determining what has been tampered with as not forged can produce serious problems. From this viewpoint, the fact that double-compressed frames are classified as forged frames does not cause serious problems.

7. Conclusions

In this paper, we introduced an object-based video-tampering detection network using symmetrically overlapped motion residual features. Three motion residuals with overlapped and different temporal windows were created to enhance the discriminability of video frames. After forming patches on the proposed motion residuals, these features were fed as input to the basic CNN. We trained two networks of two different pairs. The first network trained on a pair consisting of forged and non-forged frames to determine whether a frame has been tampered with. The second network trained on a pair of pristine and double-compressed frames in combination with the first network to solve the three-class identification, which classifies a frame into pristine, double-compressed, and forged. The simulation results revealed that the proposed video-frame identification produced more accurate performance in both two- and three-class identifications for various metrics.

Author Contributions: T.H.K. proposed the framework of this work, carried out the whole experiments, and drafted the manuscript. C.W.P. created the input dataset and helped with the experiment. I.K.E. initiated the main algorithm of this work, supervised the whole work, and wrote the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (NRF-2018R1D1A1B07046213).

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

References

1. Alsmirat, M.A.; Al-Hussien, R.A.; Al-Sarayrah, W.T.; Jararweh, Y.; Etier, M. Digital video forensics: A comprehensive survey. *Int. J. Adv. Intell. Paradig.* **2020**, *15*, 437–456. [CrossRef]
2. Javed, A.R.; Jalil, Z.; Zehra, W.; Gadekallu, T.R.; Suh, D.Y.; Piran, M.J. A comprehensive survey on digital video forensics: Taxonomy, challenges, and future directions. *Eng. Appl. Artif. Intell.* **2021**, *106*, 104456. [CrossRef]

3. Wang, W.; Farid, H. Exposing digital forgeries in video by detecting double MPEG compression. In Proceedings of the 8th Workshop on Multimedia and Security, Geneva, Switzerland, 26 September 2006; pp. 37–47. [CrossRef]
4. Vazquez-Padin, D.; Fontani, M.; Bianchi, T.; Comesana, P.; Piva, A.; Barni, M. Detection of video double encoding with GOP size estimation. In Proceedings of the IEEE International Workshop on Information Forensics and Security, Costa Adeje, Spain, 2–5 December 2012; pp. 151–156. [CrossRef]
5. Milani, S.; Bestagini, P.; Tagliasacchi, M.; Tubaro, S. Multiple compression detection for video sequences. In Proceedings of the IEEE 14th International Workshop on Multimedia Signal Processing, Banff, AB, Canada, 17–19 September 2012; pp. 112–117. [CrossRef]
6. Jiang, X.; Wang, W.; Sun, T.; Shi, Y.Q.; Wang, S. Detection of double compression in MPEG-4 videos based on Markov statistics. *IEEE Signal Process. Lett.* **2013**, *20*, 447–450. [CrossRef]
7. Stamm, M.C.; Lin, W.S.; Liu, K.R. Temporal forensics and anti-forensics for motion compensated video. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 1315–1329. [CrossRef]
8. Gironi, A.; Fontani, M.; Bianchi, T.; Piva, A.; Barni, M. A video forensic technique for detecting frame deletion and insertion. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 6226–6230. [CrossRef]
9. Yang, J.; Huang, T.; Su, L. Using similarity analysis to detect frame duplication forgery in videos. *Multimed. Tools Appl.* **2016**, *75*, 1793–1811. [CrossRef]
10. Vaishnavi, D.; Subashini, T.S. Application of local invariant symmetry features to detect and localize image copy move forgeries. *J. Inf. Secur. Appl.* **2019**, *44*, 23–31. [CrossRef]
11. Li, Y.; Zhou, J. Fast and effective image copy-move forgery detection via hierarchical feature point matching. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 1307–1322. [CrossRef]
12. Park, J.Y.; Kang, T.A.; Moon, Y.H.; Eom, I.K. Copy-move forgery detection using scale invariant feature and reduced local binary pattern histogram. *Symmetry* **2020**, *12*, 492. [CrossRef]
13. Ricaho, C.; Gaobo, Y.; Ningbo, Z. Detection of object-based manipulation by the statistical features of object contour. *Forensic Sci. Int.* **2014**, *236*, 164–169. [CrossRef]
14. 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, Hong Kong, China, 16–19 December 2015; pp. 719–722. [CrossRef]
15. Kodovsky, J.; Fridrich, J. Calibration revisited. In Proceedings of the 11th ACM workshop on Multimedia and security, Princeton, NJ, USA, 7–8 September 2009; pp. 63–74. [CrossRef]
16. Kodovsky, J.; Fridrich, J. Ensemble classifiers for steganalysis of digital media. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 432–444. [CrossRef]
17. 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. [CrossRef]
18. Penvy, 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, 2 March 2007; p. 650506. [CrossRef]
19. Saddique, M.; Asghar, K.; Bajwa, U.I.; Hussain, M.; Habib, Z. Spatial video forgery detection and localization using texture analysis of consecutive frames. *Adv. Electr. Comput. Eng.* **2019**, *19*, 97–108. [CrossRef]
20. Sitara, K.; Mehtre, B.M. Differentiating synthetic and optical zooming for passive video forgery detection: An anti-forensic perspective. *Digit. Investig.* **2019**, *30*, 1–11. [CrossRef]
21. Aloraini, M.; Sharifzadeh, M.; Schonfeld, D. Sequential and patch analyses for object removal video forgery detection and localization. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 917–930. [CrossRef]
22. Doulamis, N.; Doulamis, A.; Protopapadakis, E. Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, *2018*, 7068349. [CrossRef]
23. Yao, Y.; Shi, Y.; Weng, S.; Guan, B. Deep learning for detection of object-based forgery in advanced video. *Symmetry* **2018**, *10*, 3. [CrossRef]
24. Kohli, A.; Gupta, A.; Singhal, D. CNN based localisation of forged region in object-based forgery for HD videos. *IET Image Process.* **2020**, *14*, 947–958. [CrossRef]
25. Yang, Q.; Yu, D.; Zhang, Z.; Yao, Y.; Chen, L. Spatiotemporal trident networks: Detection and localization of object removal tampering in video passive forensics. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *31*, 4131–4144. [CrossRef]
26. 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; p. 94090J. [CrossRef]
27. Fridrich, J.; Kodovsky, J. Rich models for steganalysis of digital images. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 868–882. [CrossRef]
28. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
29. Qadir, G.; Yahaya, S.; Ho, A.T.S. Surrey university library for forensic analysis (SULFA) of video content. In Proceedings of the IET Conference on Image Processing, London, UK, 3–4 July 2012; p. 94090J. [CrossRef]
30. Bestagini, P.; Milani, S.; Tagliasacchi, M.; Tubaro, S. Local tampering detection in video sequences. In Proceedings of the 2013 IEEE 15th International Workshop on Multimedia Signal Processing, Pula, Italy, 30 September–2 October 2013; pp. 488–493. [CrossRef]

31. Aloraini, M.; Sharifzadeh, M.; Agarwal, C.; Schonfeld, D. Statistical sequential analysis for object-based video forgery detection. *Electron. Imag.* **2019**, *2019*, 543-1–543-7. [CrossRef]
32. Pevny, T.; Bas, P.; Fridrich, J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 215–224. [CrossRef]
33. 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. [CrossRef]
34. 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. [CrossRef]