Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

Video forgery detection method based on local difference binary

Yerel fark ikililerine dayalı video sahtecilik tespit yöntemi

Guzin ULUTAS[1], *Beste USTUBIOGLU*[2*], *Mustafa ULUTAS*[3], *Vasif NABIYEV*[4]

[1,2,3,4] Department of Computer Engineering, Faculty of Engineering, Karadeniz Technical University, Trabzon, Turkey.
guzin@ieee.org, bustubioglu@ktu.edu.tr, ulutas@ieee.org, vasif@ktu.edu.tr

Received/Geliş Tarihi: 07.05.2019 Revision/Düzeltme Tarihi: 03.01.2020 doi: 10.5505/pajes.2020.75768
Accepted/Kabul Tarihi: 23.01.2020 Research Article/Araştırma Makalesi

Abstract

Recently, the rapid development of video editing software has made video forgery applicable. Researchers have proposed methods to detect forged video frames. These methods utilize codec properties, motion artifacts, noise effect and frame similarity to detect forgery. Execution time and low detection accuracy are the two main drawbacks of forgery detection methods reported in the literature. In this study, a new frame duplication detection method using Local Difference Binary (LDB) is proposed to extract features from the frames. Distance between similar frames that have similar feature vectors are is used by the method to estimate Distance of Forgery and to determine the exact location of duplicated frames. PSNR between similar frames are is then used to group them into three classes, and rule-based mechanism reports forged frames according to the membership to classes. Experimental results indicate that the proposed method has lower execution time with higher accuracy than similar works.

Keywords: Video forgery, LDB, Frame duplication detection, Distance of forgery.

Öz

Son yıllarda, video düzenleme yazılımının hızlı gelişimi video sahteciliğini uygulanabilir hale getirmiştir. Araştırmacılar sahte video karelerini tespit etmek için yöntemler önermiştir. Bu yöntemler sahteciliği tespit etmek için kodek özelliklerini, hareket artefaktlarını, gürültü etkisini ve çerçeve benzerliğini kullanmaktadır. Literatürde sahtecilik tespiti için önerilen yöntemlerin iki ana dezavantajı çalışma zamanı ve düşük tespit doğruluğudur. Bu çalışmada, çerçevelerden özellikler çıkarmak için Yerel Fark İkililerini (LDB) kullanan yeni bir çerçeve tekrarlama tespit yöntemi önerilmiştir. Benzer özellik vektörlerine sahip benzer çerçeveler arasındaki mesafe, Sahtecilik Mesafesini tahmin etmek ve kopyalanan çerçevelerin tam yerini belirlemek için kullanılmaktadır. Benzer çerçeveler arasındaki PSNR daha sonra bunları üç sınıfa gruplamak için kullanılır ve kural tabanlı mekanizma sahte çerçeveleri sınıf üyeliklerine göre raporlar. Deneysel sonuçlar, önerilen yöntemin, benzer çalışmalara kıyasla daha yüksek hassasiyetle daha düşük uygulama süresine sahip olduğunu göstermektedir.

Anahtar kelimeler: Video sahteciliği, LDB, Çerçeve tekrarlama tespiti, Sahtecilik uzaklığı.

1 Introduction

In recent years, anti-forensic researches on multimedia files (such as image, video, etc.) gains popularity in literature. Two factors are effective on increasing attention: Widespread usage of multimedia files in daily life and development of easily usable multimedia editing tools. Multimedia files can be captured by image acquisition tools such as camcorder, cell phone, etc. at anytime and anywhere and they can be used for various purposes such as for diagnostic-purposes in medical systems, as an evidence in courts. Easily usable multimedia editing tools makes easy to modify the content of a multimedia file for malicious intentions. Thus, a new problem was raised with an increase in widespread usage of multimedia acquisition tools and development of easily usable multimedia editing tools: Authenticity of the multimedia files.

Two approaches are used in the literature for ensuring the authenticity of the multimedia files: Active and Passive methods. The former method constructs the specially created information called watermark and embeds it into the multimedia files using special techniques. However, active methods necessitate usage of specially written software to embed the watermark information into the image, or specially equipped hardware must be used during the capturing of multimedia files. Researchers suggested methods that fall into the second category to authenticate multimedia files because using software or hardware before image acquisition is somehow troublesome. Passive methods do not necessitate any priori information to authenticate the multimedia files and they use statistical properties of the files to determine the forgery. Recently, various passive methods have been proposed to deal with image and video authentication problem. In this work, we are especially interested of video authentication problem because less effort has been made in that field compared to image authentication area.

Wang et al. have suggested the first passive method for video authentication in 2006 [1]. Their technique uses ~~an~~ evidence to decide the forgery. Doubly compressed MPEG video sequence introduces specific artifacts and their absence in a video designates a forgery operation. After this work, in 2007 Wang et al. suggested another technique [2]. In their method, the forged video is divided into subsequences and is calculated correlation matrix for each overlapping subsequences is calculated. The correlation coefficient value between two matrixes gives a clue about the similarity of corresponding subsequences. If the correlation coefficient is larger than a predefined threshold value, the algorithm divides the frames into non-overlapping blocks and consults the similarity of the corresponding blocks in two sequences. Another technique proposed by Wang et al. detected traces of forgery in de-interlaced and interlaced videos [3]. Their method shows that forgery operation disturbs correlations introduced by the

[*] Corresponding author/Yazışılan Yazar

camera or software de-interlacing algorithms. It also uses that the motion between fields of a single frame and across fields of adjacent frames should be equal for interlaced videos. In 2008, Luo et al. explored the temporal patterns of the blocking artifacts in video sequences [4]. Their method showed that various types of frames contain different block artifacts in MPEG compression and a group of pictures (GOP) has a regular pattern. Recompression after the forgery operation that removes some frames from the original MPEG video file affects the block artifact strength of the recompressed video, and the method uses this artifact to detect forgery operation. Wang et al. consulted a double quantization effect in the test video to determine the forgery operation [5]. Their study also uses attributes of MPEG standards to determine the forgery operation. In 2009, Su et al. detected frame deletion forgery in MPEG files [6]. Motion-compensated edge artifacts are used to determine the modification on correlation between adjacent frames. Break point indicates the point where frame deletion occurs. In the same year, Zhang et al. exploited Ghost shadow artifact to detect removed objects from a video by inpainting operations [7]. Their method segments each frame into static background and moving foreground and computes optical flow to create foreground music. Accumulative differences between frames are also used to create moving foreground track. If foreground track does not consistent with foreground mosaic, the method decides the forgery. Hu et al. obtained temporally informative representative images from the subsequences and used their DCT coefficients for fingerprint generation [8]. Their results show that the algorithm is robust against MPEG compression. Lin et al. utilized spatial and temporal similarity to detect duplicated subsequences and firstly extracted candidate clips that give similar histograms [9]. The spatial similarity is then consulted to determine the exact location of forgery operation. Sun et al. used MPEG double compression traces to detect forgery [10]. The method obtained feature vector of size 1x12 from each group of pictures (GoP) and adopted machine-learning framework to improve the detection accuracy. In 2012, Subramanyam et al. proposed a video forgery detection method using Histogram of Gradients (HoG) features [11]. The authors prefer to use HoG as feature extraction method due to its robustness against various signal-processing manipulations. Their method also utilized MPEG properties to detect temporal similarity. Kancherla et al. applied Markov Models to motion in videos to detect video forgery. Their method emphasized that motion information is extracted from the video by applying collusion on successive frames that gives base frame. The algorithm extracts motion frame by subtracting actual frame and base frame. Markov model is used to model the motion. When pattern recognition is applied to the extracted features, the algorithm decides the forgery operation. In 2013, Chao et al. used optical flow consistency to detect forgery operation [12]. Optical flow is generated by the method, and the type of the forgery is determined (Frame deletion or frame insertion). The method applies two different algorithms to detect forged sequences according to the forgery type. Lin et al. determined temporal similarity using histogram difference of two adjacent frames in the RGB color space [13]. If temporal similarity exists between two subsequences in a test video, the method calculates spatial similarity between corresponding frames of subsequences. A classifier is constructed to label the videos as forged or not according to the results of spatial and temporal similarity. Liao et al. extracted Tamura texture features from each frame of video and an eigenvector matrix is created using these features [14]. The matrix is lexicographically sorted, and vectors in each row are compared to determine the forged sequences. Lin et al. proposed a method for the detection of region-level forgery from test videos [15]. Their method investigated two inpainting operations: temporal copy paste and exemplar-based texture synthesis. Spatio temporal coherence analysis, tampered slice detection and region localization are realized by the method. Su et al. calculated the features of difference between frames by using k-Singular Value Decomposition (kSVD) [16]. Features are transformed into smaller space with random projection and then the features are clustered using k-means. The final result denotes the detection result. In 2014, Yang et al. proposed a similarity-analysis based method for frame duplication detection [17]. SVD is applied to each frame and features are obtained. Euclidean distance is measured between the features of each frame and the reference frame. Similarities between the subsequence of features denote the forgery operation. A finer analysis is applied to the candidate subsequences via block analysis to detect the exact location of forgery. Singh et al. proposed a passive method with two different algorithms to detect frame and region duplication forgeries in videos [23]. The algorithm I of proposed method detects frame duplication forgery in videos by obtaining the mean features of each video frame for evaluating the correlation between sequences. The algorithm II detects these region duplication forgeries in videos by locating the position of error with threshold process. In 2017, Ulutas et al. used binary features to detect frame mirroring and frame duplication forgery [24]. The method extracts binary features from frames and determines the similarity among features. The same authors also proposed another study based on BoW model to detect frame duplication forgery in 2018 [25]. Their method uses BoW to create visual words and build a dictionary from Scale Independent Feature Transform (SIFT) keypoints of frames in video.

In recent years, some works also consider deep learning techniques to detect object based forgeries on the videos [26]-[30]. [26] utilized from CNN based deep learning approach to detect object based forgery. However, forgery process considered by their work does not use frame duplication or insertion technique. Their technique considers object based forgery. In 2018, Bakas et al. proposed a deep learning architecture which utilizes artifacts in the I-frames to detect double quantization. They used TRACE library for their comparisons [27]. [28] constructs their method using I3D and Siamens network to detect video forgery operation. Their method implements coarse to fine approach to detect forged sequences. Frame and video level forgery detection are realized by their method. In 2019, Raveendra et al. Detected double compression artifacts by adapting Markov based features [29]. Gabor features are then used for forgery detection as a feature for deep neural network. They construct a dataset to show the effectiveness of their method. D'Avino et al. performed video forgery detection using deep learning with an architecture based on recurrent neural networks and auto encoder [30]. Autoencoder learns model of the source using a few pristine frames. If the material does not fit the learned model, the method classifies it as forged video.

While some methods reported above are using codec characteristics of the video [1],[3]-[5],[10]-[11], some of them assume that the malicious user modifies the motion in the video and motion analysis can be used to find the trace of forgery [6]-[7],[12],[15],[16]. The other methods given in [2],[8],[9],[13],[14],[17] extract features from the frames and search subsequences that give similar features. The methods in the last category are independent of video codec properties and they don't use motion artifacts as a clue for forgery determination.

In this work, we proposed a new codec and motion independent frame duplication detection method. The main motivation of the method is to ensure improved accuracy with less execution time. We used a new binary descriptor proposed by Yang et al. in 2014 called by Local Difference Binary (LDB) to extract features from the frames [18]. LDB achieves similar computational speed and robustness as state-of-the-art binary descriptors with higher distinctiveness as stated by the authors. LDB feature vectors that are extracted from the frames are compared to determine the similar frames and then a new method called by Distance of Forgery is applied on ~~the~~ similar frames to decide the exact distance between the replicated subsequences. Frame pairs obtained from these two steps (feature extraction and the determination of exact distance) are the candidate pairs. In the last step, the method clusters the pairs into three groups (highly similar, similar, less similar) according to PSNR value between them. The method decides the forged sequences on the test video using the number of elements in the clusters.

Similar works in the literature [2],[13],[17],[23],[24],[25] are realized to make a fair comparison with the proposed method. Experimental results show that the proposed method gives better Detection Accuracy, Precision Rate and Recall Rate compared to similar works [2],[13],[17],[23]-[25]. When execution time is considered, the method determines the forged sequences faster than the others [2],[13],[17],[23],[24,[25].

The rest of the paper is organized as follows. Section 2 defines the method to extract features from the frames, LDB. The details of the method and experimental results are given in Section 3 and Section 4 respectively. Some conclusions are also drawn in the last section.

2 Local difference binary

Yang et al. introduced a new binary descriptor called LDB in 2014 [18]. This new feature description technique ensures higher distinctiveness compared to similar binary feature extraction techniques [19]-[22]. LDB divides the image into grids, and use average pixel values of the grids and first-order gradients to generate descriptor.

Assume that the image *I* is divided into *nxn* equal-sized grids*.* The feature extraction technique extracts information from each cell and applies binary tests on them to obtain representative feature. Let *F* denotes the function that is used for information extraction from the cells. The equation given in (1) shows the binary test τ and *i, j* denote the cells in the current image. It gets two values and compares their values to generate binary information.

$$\tau\big(F(i),F(j)\big)=\begin{cases}1, if\ \Big(\big(F(i)-F(j)\big)>0\Big)\wedge i\neq j\\ 0, \qquad otherwise\end{cases} \qquad (1)$$

The F is determined in [18] as the average function due to its computational speed. The average function is applied to each cell to extract information. However, the authors emphasized that using average pixel value for representative purposes is too coarse approach. First-order gradients of image *I* are also evaluated to improve the resiliency of the feature extraction technique. Function *F* returns three results for a cell denoted by i using (2) where $I(i,k)$ denotes the kth pixel of ith cell and m represents the number of pixels in a cell.

$$F=\left\{\frac{1}{m}\sum_{k=1}^{m}I(i,k)\,,Gradient_x(i),Gradient_y(i)\right\} \qquad (2)$$

The first result of *F* is the average intensity pixel value of *i*th cell. The last two results denoted by $Gradient_x(i)$ and $Gradient_y(i)$ are the average values in the regional gradients of grid cell *i* in the *x* and *y* directions respectively.

The Feature extraction algorithm obtains three results for each cell and performs binary comparison given in (1) on pairwise grid cells to compare the corresponding results. LDB descriptor is constructed with $3n^2(n^2-1)/2$ (that is the total number of comparisons) binary values.

Choosing the best grid size used for feature extraction is another problem. If the size of the cells is selected to be small, the descriptor's stability would be lower however it can capture more details. Otherwise, the descriptor would be more stable however it was coarser. The authors proposed to combine the results of multiple-gridding choices. For example, if an image is divided into 2x2 and 3x3 cells, binary results obtained from them are combined to form the descriptor.

In this work, we used 2x2, 3x3 and 4x4 to form the cells and combine the binary results to create a feature vector for each frame of size 1x486 bits. 18 bits are obtained from pairwise comparison of 2x2 cells and 108, 360 bits are calculated from 3x3 and 4x4 cells respectively.

3 Proposed method

In this section we give the details of the proposed video forgery detection method, which detects duplicated frames in the forged video. Figure 1 shows an example of frame duplication forgery.

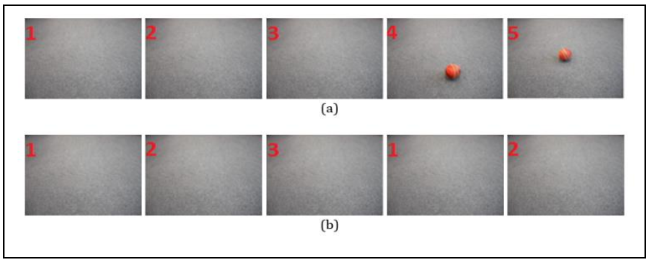

Figure 1. Video frame duplication forgery example.

First two frames of the original video are copied and pasted onto frames 4 and 5, as can be seen in figure. The ball in the scene will disappear due to the frame duplication forgery. A general framework of the algorithm is also given in Figure 2.

The algorithm consists of four parts: Feature Extraction from the frames, Determination of the Distance of Forgery, Grouping the similar frames. The algorithm firstly divides the video into frames and extracts features from the frames using LDB. Feature vectors are used to determine similar frames. The distance between the copied and replicated sequences are then estimated by using the *Distance of Forgery* method that uses the list of similar frames. This method gets the similar frames as input and decides the distance between the copied and replicated parts. Similar frame pairs that violate the determined distance are extracted from the similar frames list. In the last step, the algorithm groups the similar frame pairs into three classes (highly similar, similar, less similar) according to their PSNR values. The algorithm decides the forged sequences using the member of classes. The algorithm will be explained in details as below.

Feature Extraction from the frames using LDB

The method divides the video into frames to extract features from them as the first step. LDB is used to extract binary information from the frames and then similar frames are determined from their corresponding feature vectors by calculating Hamming distance. Assume that input video with N frame is denoted by $V = \{V^i | i = 1 \cdots N\}$. The method calculates the gradients of V^i in x and y directions, G_x^i and G_y^i. The frame and its gradients are divided into 2x2, 3x3 and 4x4 cells respectively and average values are calculated from each cell.

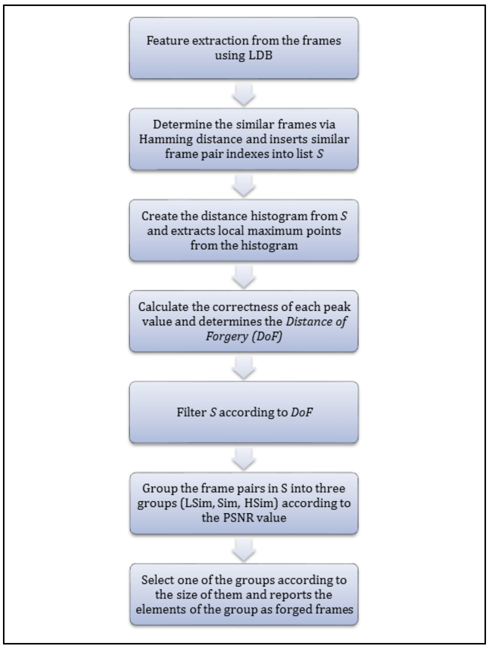
Feature extraction from the frames using LDB
Determine the similar frames via Hamming distance and inserts similar frame pair indexes into list *S*
Create the distance histogram from *S* and extracts local maximum points from the histogram
Calculate the correctness of each peak value and determines the *Distance of Forgery (DoF)*
Filter *S* according to *DoF*
Group the frame pairs in S into three groups (LSim, Sim, HSim) according to the PSNR value
Select one of the groups according to the size of them and reports the elements of the group as forged frames

Figure 2. General flow diagram of the proposed method.

The total of 18-bit information is obtained from binary comparisons on 2x2 cells on the current frame and its gradients. Figure 3 shows the graphical demonstration of extracting the binary information from 2x2, 3x3 and 4x4 cells. The algorithm also divides the frame and its gradients into 3x3 and 4x4 cells and calculates average intensity values from each cell and its gradients.

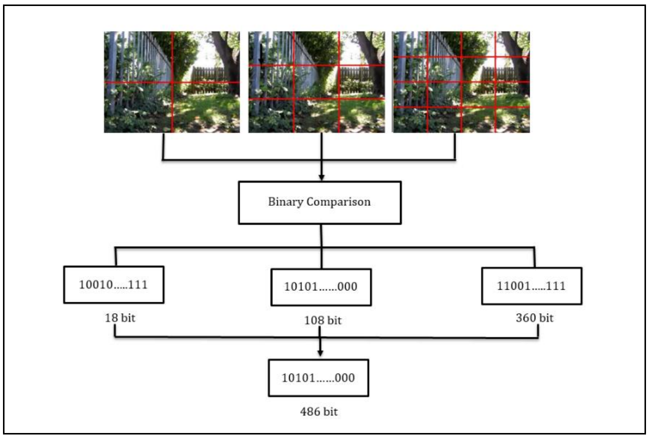

Figure 3. Graphical demonstration of feature extraction method for 2x2 cells.

When the cells in 3x3 configuration are considered, each cell is compared with the remainders. The first cell is compared with other eight cells; second cell is compared with other seven cells and so on. Thus, 108 bits are obtained from the $(8 \times 9)/2 = 36$ comparisons that are realized on V^i, G_x^i and G_y^i. 360 bits are obtained in the same manner when the cells in 4x4 configuration are considered.

As a result, a feature vector of size 1x486(18+108+360) is obtained from each frame. Assume that feature vectors that represent the corresponding frames denoted by $F = \{F^i | i = 1 \cdots N\}$. Each vector has 486 binary values and the algorithm expects duplicated frames that have similar feature vectors. Two feature vectors correspond to the same frames cannot be equal due to the compression artifacts.

Determine the Similar Frames via Hamming Distance and Inserts Similar Frame Pair Indexes into List S

The method determines similar frames using feature vectors. Each feature vector is compared with vectors, which follows it. Hamming distance is used for comparison purposes due to binary values in the vectors. Assume that the current feature vector be F^i. The vectors from F^{i+w} to F^N are tested to determine the similarity. The algorithm starts to test after w vectors because neighboring frames give similar vectors. The condition given in (3) is used to compare F^i with F^j and F_k^i denotes the kth element of ith vector. If the number of corresponding different elements does not exceed a predefined threshold value t, the algorithm assumes that two vectors are similar and inserts their index values into similar frames list S.

$$\sum_{k=1}^{486}\left(F_k^i \oplus F_k^j\right) \leqslant t \Rightarrow S_1^t = i, S_2^t = j \qquad (3)$$

In the last of this part, the list S that contains similar frame pairs are transferred to the following step to determine the distance of forgery.

Create the distance histogram from S and extracts local maximum points from the histogram

The method determines the distance between the copied and replicated sequences using the similar frames list S. The list contains two columns, which designates index values of the copied and replicated frames respectively. The method calculates the distance between the frame indexes and constructs absolute distance vector D.

$$D^t = |S_1^t - S_2^t| \qquad (4)$$

The method calculates the histogram of D to determine the frequencies of distances. Assume that histogram of D denoted by H. Local maximum values in the histogram shows the distances that are encountered frequently in the similar frames list S. These values in H are extracted and accumulated in a list $L = \{lmax_i | i \in [1 \cdots lmax_{number}]\}$ where $lmax_{number}$ denotes the number of local maximums in the current histogram. For example, if the current histogram contains four maximum values, the list will contain four elements that are encountered frequently compared to others in S.

Calculate the Correctness of Each Peak Value and Determine the Distance of Forgery

Each local maximum point is evaluated to determine the accuracy of it. Assume that the current maximum point will be evaluated be $lmax_i$. Similar frame pair indexes are extracted from S such that the distance between them is equal to $lmax_i$.

The equation given in (5) is used to filter the frame pairs from *D.*

$$D^t = lmaxi \Rightarrow temp_1^t = S_1^t, temp_2^t = S_1^t \tag{5}$$

The list *temp* contains indexes of frames such that their distance is equal to $lmax_i$. The method calculates first derivative of *temp* along the first column and second column respectively and obtains *difx₁* and *difx₂* vectors. If a forgery operation has been occurred in the test video, index values of similar frames must be consecutive. For example, if $lmax_i = 40$ and the frames between 10-30 are copied and pasted onto the 50-70, the list *temp* must contain frame pairs that are consecutive such as $temp = \{1050,1151,1252,1353,1454,\cdots,3070\}$. Thus, *difx₁* and *difx₂* are obtained to be $\{111\cdots1\}$ and $\{111\cdots1\}$ respectively. However, the values in *temp* cannot contain correct results all the time due to the compression artifacts and some frame pairs cannot be detected by the algorithm in a consecutive manner such as $temp = \{1050,1252,1353,1555,\cdots,3070\}$. In this case the first derivative of the first and second columns will be $\{212\cdots1\}$ and $\{212\cdots1\}$.

The algorithm as seen in the Figure 4 evaluates the first derivatives and decides the correctness of the $lmax_i$. A window of size 1x*w* is constructed using 1-values and is slided onto the *difx₁* and *difx₂* to determine the correctness of the peak value. Euclidean distance between the window and the *difx₁* and *difx₂* are calculated at each step separately and inserted into *fx₁* and *fx₂* respectively. If the peak value corresponds to the *distance of forgery,* the window will correlate the derivatives and elements of *fx₁* and *fx₂* will be smaller than a threshold value *th*. Otherwise, *fx₁* and *fx₂* contain elements that are larger than *th* and the algorithm ignores the peak value.

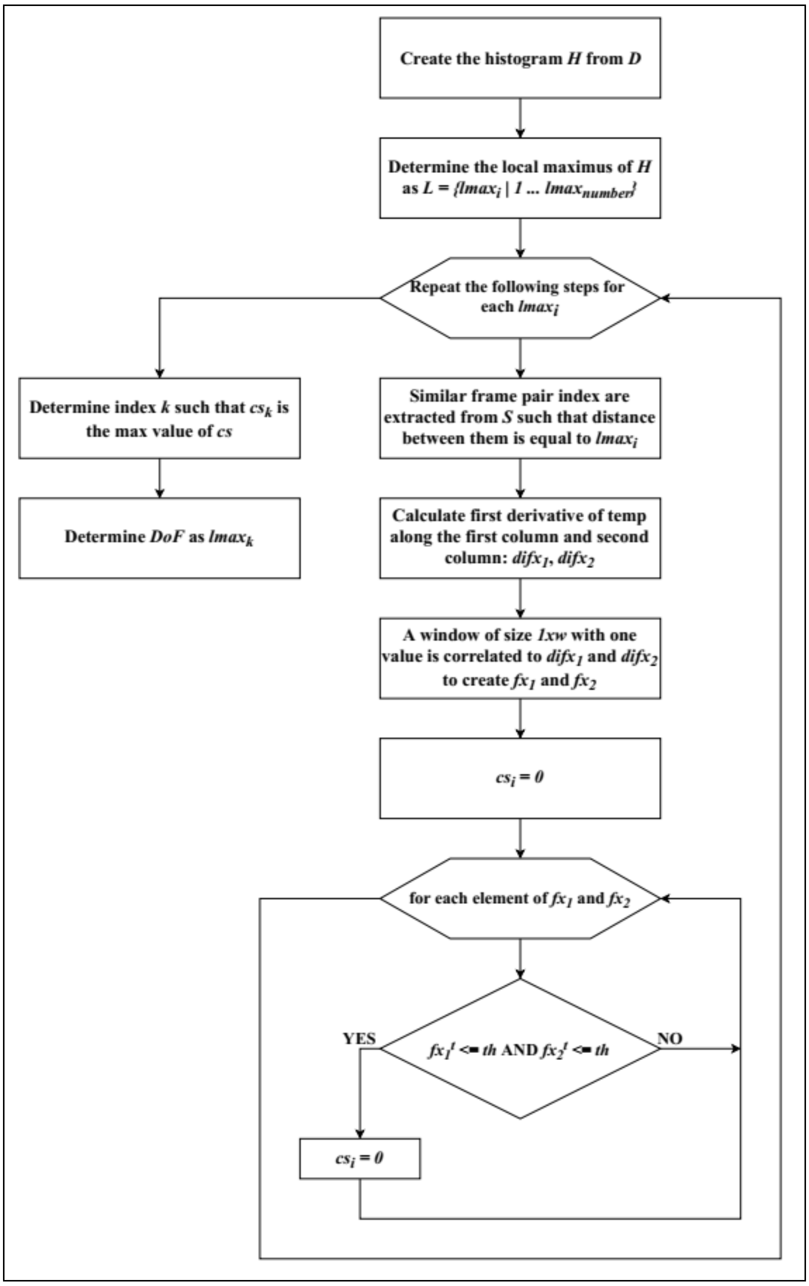
Create the histogram H from D
Determine the local maximus of H as L = {lmax_i | 1 ... lmax_number}
Repeat the following steps for each lmax_i
Determine index k such that cs_k is the max value of cs
Determine DoF as lmax_k
Similar frame pair index are extracted from S such that distance between them is equal to lmax_i
Calculate first derivative of temp along the first column and second column: difx_1, difx_2
A window of size 1xw with one value is correlated to difx_1 and difx_2 to create fx_1 and fx_2
cs_i = 0
for each element of fx_1 and fx_2
YES
fx_1^t <= th AND fx_2^t <= th
NO
cs_i = 0

Figure 4. General flow diagram of the *DoF.*

The equation given in (6) is applied on to the *fx₁* and *fx₂ of size 1xM* to decide the correctness of the local maximum value and the correctness score cs_i for $lmax_i$ and itself are inserted into a list *corr*.

$$cs_i = \sum_{t=1}^{M} (fx_1^t \leqslant fx_1^t) \wedge (fx_2^t \leqslant fx_1^t) \Rightarrow corr_1^t = lmaxi, corr_2^t = cs_i \qquad (6)$$

Where *corr* contains the values that give an idea about the correctness of the corresponding peak values $lmax_i$, cs_i. The steps given above are applied on the other peak values and the list *corr* will be created. Local maximum value $lmax_i$ that has the maximum correctness score cs_i, is chosen to be *Distance of Forgery, DoF.*

Filter S according to DoF

The proposed method filters the similar frames list *S* with *DoF* using (7) to create modified list mod*S*. The following subsection uses *modS* as the input to determine the exact location of forgery.

$$D_i = DoF \Rightarrow modS_1^i = S_1^i, modS_2^i = S_2^i \qquad (7)$$

Group the frame pairs in modS into three groups according to the PSNR value

In this part of the algorithm, similar frame list *modS* is grouped into three sections: Highly similar frames, similar frames and less similar frames. Peak to Signal Noise Ratio (PSNR) is used to group the frame pairs. PSNR gives an idea about the similarity of two images. The method uses the following ranges for grouping purposes. *Less similar, Similar and Highly Similar* as given in (8) lists contain frame pairs according to their similarity.

$$\begin{gathered}(t_1 \leq PSNR < t_2) \rightarrow Lesssimilar \\ (t_2 \leq PSNR < t_3) \rightarrow Similar \\ (PSNR \geq t_3) \rightarrow Highlysimilar\end{gathered} \qquad (8)$$

PSNR is calculated for each similar frame pairs in *modS* and they are grouped according to the PSNR. Assume that the lists denoted by *LSim, Sim* and *HSim* contain frame pairs after grouping operation and $Size(LSim)$ gives the number of frame pairs in the current list. If the third group *HSim* contains similar frames, the method determines the frame pairs in this group as forged frames. Otherwise, the method signs the frame pairs in that group (*Sim* or *LSim*) which has more elements as forged.

The method can be given in the form of steps as follow.

The general outline of the proposed algorithm

- ❖ Divide the video into frames,
- ❖ Extract corresponding feature vectors from each frame,
- ❖ Compare the feature vectors using (3) and inserts similar pairs into list *S*,
- ❖ Constructs distance vector D using (4),
- ❖ Find local maximum points of D and inserts them into L.
- ❖ Repeat the following steps for $i \in [1 \cdots lmax_{number}]$
    - ➢ temp ← Extracts the frame pairs which their distance values equal to $lmax_i$ from *S*,
    - ➢ difx₁ ← Calculates the first derivative of the first column of *temp*,
    - ➢ difx₂ ← Calculates the first derivative of the second column of *temp*,
    - ➢ Slides a window of size 1x*w* with 1-values over *difx₁* and calculates Euclidean distance at each step. Inserts Euclidean distance results to a list *fx₁*,
    - ➢ Slides a window of size 1x*w* with 1-values over *difx₂* and calculates Euclidean distance at each step. Inserts Euclidean distance results to a list *fx₂*,
    - ➢ Determine the correctness score (cs_i) for the current peak value $lmax_i$ using (6) and embeds the $lmax_i$ and their correctness value cs_i into list *corr*,
- ❖ Determine *DoF* as the point $lmax_i$, which has maximum correctness score as in,
- ❖ Filter *S* using *DoF* as in (7),
- ❖ Calculate PSNR values between similar frame pairs and then grouped them according to their PSNR values. The lists *LSim, Sim* and *HSim* contain frame pairs that are fall into the less similar, similar and highly similar groups according to the given ranges.
    - ➢ If $Size(HSim) > 0$ The method determines the frames in this list as forged,
    - ➢ Else If $Size(Sim > LSim)$,
        - ▪ The method determines the frames in *Sim* list as forge
    - ➢ Else
        - ▪ The method determines the frames in *LSim* list as forged.

4 Experimental Results

In this section, the results of the proposed method on a number of frame duplicated forged videos are given. The experiments are implemented with 2.4 GHz dual-core i7 processor running Matlab R2014b. Tests are performed on a set of forged videos created by Virtual Dub, an open-source video editor. Test videos were downloaded from SULFA-Surrey University Library for Forensic Analysis [15], to create forged samples. Each video is approximately 10 seconds long with resolution of 320×240 and 30 frames per second. All videos have been shot after carefully considering both temporal and spatial video characteristics.

Videos, can_220_book, can_220_flap(1), can_220_flap(2), can_220_garden(1), can_220_garden(4), can_220_man(2), can_220_road(1), can_220_room(3), can_220_street(3), fuji_2800_man (2), fuji_2800_outdoor(4), fuji_2800_road(2), fuji_2800_road(5),fuji_2800_stair_outdoori,can_220_hallway(2),fuji_2800_busstop(4),nik_s3000_ball,nik_s3000_bridge(1),nik_s3000_indoor_stairs are used to create the test video database. In order to compare proposed method with other studies, these videos were selected from the videos used by other studies in the literature. 23 forged test videos are created and used during the tests and the details are given in Table 1.

Test database can be available at http://ceng2.ktu.edu.tr/%7Egulutas/test_database.rar.

Table 1. Test Videos.

Test video	Original	Tampered	Forgery operation
Vid. 1	407	457	170-220 are copied behind 10
Vid. 2	369	429	100-160 are copied behind 315
Vid. 3	332	372	60-100 are copied behind 280
Vid.4	353	403	200-250 are copied behind 10
Vid. 5	361	401	100-140 are copied behind 180
Vid. 6	390	440	110-160 are copied behind 10
Vid. 7	301	371	150-220 are copied behind 240
Vid. 8	436	476	220-260 are copied behind 130
Vid. 9	270	310	30-70 are copied behind 130
Vid. 10	421	481	85-145 are copied behind 220
Vid. 11	380	430	110-160 are copied behind 20
Vid. 12	310	350	30-70 are copied behind 130
Vid. 13	310	340	179-209 are copied behind 220
Vid. 14	310	340	2-32 are copied behind 70
Vid. 15	380	430	5-55 are copied behind 187
Vid. 16	460	500	223-263 are copied behind 60
Vid. 17	370	410	40-80 are copied behind 215
Vid. 18	142	182	223-263 are copied behind 60
Vid. 19	355	410	5-60 are copied behind 190
Vid. 20	363	400	138-175 are copied behind 5
Vid. 21	344	394	120-170 are copied behind 220
Vid. 22	293	323	250-280 are copied behind 10
Vid. 23	188	228	10-50 are copied behind 105

Forged videos were created using at least 30 frames to present scenarios that cannot be noticed by the human eye. Even if this is not taken into consideration, the proposed method will detect frame duplication forgery with less than 30 frames. Because in the proposed method, the features are extracted from frames, not from frame groups.

Threshold values have been set as t=5, th=2, w=10, t_1=30, t_2=34, t_3=37 during the experiments. However, threshold value t showing the number of different elements which decides the similarity of the feature vectors is important on the detection ability of the method. It is for this reason that we carried out an experiment to select the best threshold value before the comparison tests. The value of t is varied in range [5, 10, 15, 20, 486] and PR and RRs are obtained for the test videos, as given in Table 1. Figure 5 indicates that the lowest value for t is 5 which gives significiantly better PR and RRs. Therefore, we select t to be 5 for the video data set and the results obtained in the comparison tests are calculated with this value of threshold.

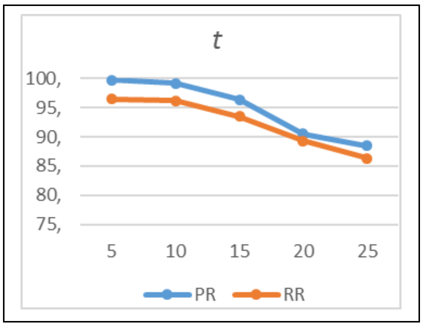

Figure 5. Choosing the best threshold value, *t*.

The details of the forgery operations on the test videos are listed in Table 1. For instance, in Vid. 2 the frames between 100 and 160 are copied and inserted after the 315th frame to generate the forged video. While the length of the original video is 407, the total length becomes 457 after forgery operation. Forged videos are then encoded with open source MPEG-4 (Level 5 Advanced Simple Profile, ASP@L5) algorithm of Xvid codec after the forgery with an open-source video editing tool, Virtual Dub.

We test the effectiveness of the proposed frame duplication detection technique using three metrics reported in the literature: Precision rate (PR), Recall Rate (RR) and Detection Accuracy (DA). These metrics give an idea about the detection capability of the proposed method and the definitions for them are given in (9) respectively.

$$\begin{aligned} PR &= \frac{TP}{TP+FP} \\ RR &= \frac{TP}{TP+FN} \\ DA &= \frac{TP+TN}{TP+TN+FP+FN} \end{aligned} \tag{9}$$

Where FN, FP, TN and TP denote “forged is detected as authentic”, “authentic is detected as forged”, “forged is detected as forged” and “authentic is detected as authentic” respectively. The total number of detections is given by *TP+TN* and the total number of frames is calculated by FN+FP+TN+TP. Thus, *DA* gives a clue about the detection performance of the proposed method. If FP increases, PR decreases and if FN increases RR decreases. Thus, False alarm ratio of the algorithm is tested using PR and RR. Higher PR and RR values indicate that the algorithm generates fewer false alarms.

The First experiment tests the proposed method on the forged videos given in Table 1 and calculates the average PR, RR and DA values. Table 2 also shows the detailed PR, RR and DA values of the proposed method for each test videos.

The method is also compared to similar works in the literature [2],[13],[17],[23]-[25] to show effectiveness. This works use SULFA database and ınternet videos. Therefore, we also recoded this works for testing their results on our dataset to make a fair comparison. Figure 6 indicates the average *PR* values of the methods when the test videos are used for testing purposes. The method gives higher PR value compared to the method in [2],[13],[17],[23]-[25]. When RR values are evaluated, the method has the highest RR value after [25] as can be seen in Figure 6. Higher RR value indicates that the method detects forged frames with more accuracy. Figure 6 also shows the performance of the method when DA values are considered. DA values give an idea about the general performance of the system. While the method has approximately %96.43 DA, the others have worse overall performance as can be seen in Figure 6.

Table 2. Detailed PR, RR and DA values for the proposed method.

Test Video	PR	RR	DA
Vid. 1	1	0.99	0.99
Vid. 2	0.99	0.98	0.98
Vid. 3	1	0.99	0.99
Vid. 4	1	0.90	0.92
Vid. 5	1	0.98	0.98
Vid. 6	1	0.99	0.98
Vid. 7	1	0.97	0.98
Vid. 8	0.93	0.99	0.94
Vid. 9	0.99	0.93	0.94
Vid. 10	0.88	0.99	0.94
Vid. 11	1	0.99	0.99
Vid. 12	1	0.98	0.98
Vid. 13	1	0.95	0.94
Vid. 14	1	0.99	0.99
Vid. 15	1	0.99	0.99
Vid. 16	1	0.98	0.98
Vid. 17	1	0.98	0.98
Vid. 18	1	0.94	0.95
Vid. 19	1	0.99	0.99
Vid. 20	1	0.95	0.96
Vid. 21	1	0.9865	0.9898
Vid. 22	0.9962	0.81	0.81
Vid. 23	1	0.88	0.91

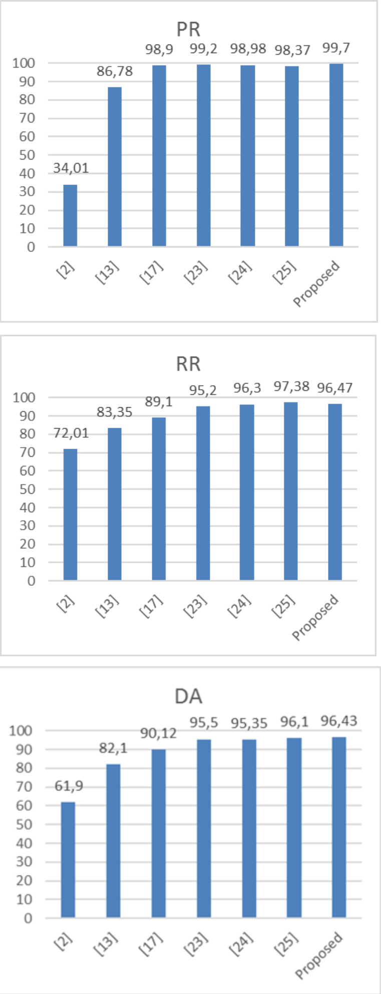

Figure 6. Performance comparisons of the proposed method.

The second experiment evaluates the execution time of the proposed method and compares it with similar works in the literature. Table 3 shows the total execution time and per frame execution time of the proposed method. When the total execution times are considered, the average execution time for the proposed method is approximately 8.5s second as can be seen in Table 3. The average execution time per frame is approximately 0.021s for the method.

Table 3: Total and per frame execution times for each test video.

Video	Execution Time	
	Total time	Time
Vid. 1	9.76	0.024
Vid. 2	10.23	0.024
Vid. 3	8.51	0.023
Vid. 4	9.25	0.023
Vid. 5	7.98	0.022
Vid. 6	8.9	0.023
Vid. 7	8.43	0.023
Vid. 8	10.355	0.024
Vid. 9	7.22	0.022
Vid. 10	9.88	0.023
Vid. 11	8.68	0.023
Vid. 12	7.13	0.023
Vid. 13	8.35	0.023
Vid. 14	8.44	0.023
Vid. 15	8.96	0.024
Vid. 16	10.95	0.024
Vid. 17	8.29	0.022
Vid. 18	3.35	0.024
Vid. 19	8.12	0.021
Vid. 20	8.06	0.022
Vid. 21	8.9	0.023
Vid. 22	7.14	0.022
Vid. 23	4.83	0.021

Per frame execution time of the proposed method is also compared with similar works and the results are reported in Table 4. All the methods referenced for comparison purposes were recoded to test their execution time on our platform. Results show that the method realizes forgery detection faster than the others. The method in [17] reports their execution time on their dataset to be 0.127s per frame. However main drawback with this method is it necessitates block based comparison between frame pairs when the source video has many salient frames. Thus, processing time of their work can increase according to structural property of the source video. We also rerun all the methods for 100 times to obtain reported average running time. All the background processes were also stopped during execution of the proposed method and the others [2],[13],[17],[23]-[25].

Table 4. Comparison of the execution times.

Method	Execution Time (s/frame)
Wang et al. [2]	294.67
Lin et al. [13]	140.6
Yang et al. [17]	0.38
Singh et al. [23]	0.024
Ulutas et al. [24]	0.01
Ulutas et al. [25]	0.2
Proposed Method	0.021

We report average execution times because difference between the execution times of the works at each run is negligible. Our method gives better execution time performance when compared to others because our technique does not necessitate block-based detection approach during forgery detection process. Main drawback with others is they have to make a finer comparison between frames especially source video consists of salient frames. However, our technique can provide better accuracy with a coarser approach.

Experiments show that, the proposed method realizes frame duplication detection with higher accuracy with less execution time compared to similar works in the literature.

5 Conclusion

Video frame duplication forgery becomes the most encountered video forgery type in recent years due to its simply implementation. Many techniques have been proposed to detect duplication forgery. Two drawbacks of the methods are the slow execution time and low RR and DA values. In this work, we proposed a new frame duplication forgery detection method with enhanced execution time and improved detection accuracy. LDB is utilized to extract features from the frames and a new method is suggested to determine the distance between the copied and pasted frames. Experimental results show that the proposed method realizes forgery detection with improved PR, RR and DA values at lower execution times compared to

6 References

[1] Wang W, Farid H. “Exposing digital forgeries in video by detecting double MPEG compression”. *Proceedings of the 8th workshop on Multimedia Security Conference*, Geneva, Switzerland, 26-27 September 2006.

[2] Wang W, Farid H. “Exposing digital forgeries in video by detecting duplication”. *Proceedings of the 9th workshop on Multimedia Security Conference*, Texas, USA, 20-21 September 2007.

[3] Wang W, Farid H. “Exposing digital forgeries in interlaced and deinterlaced video”. *IEEE Transaction on Information. Forensics and Security*, 2(3), 438-449, 2007.

[4] Luo W, Wu M, Huang J. “MPEG recompression detection based on block artifacts”. *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, San Jose, California, United States, 27-31 January 2008.

[5] Wang W, Farid H. “Exposing digital forgeries in video by detecting double quantization”. *Proceedings of the 11th ACM Workshop on Multimedia and security Conference*, Princeton, USA, 11-13 September 2009.

[6] Su Y, Zhang J, Liu J. “Exposing digital video forgery by detecting motion compensated edge artifact”. *International Conference on Computational Intelligence and Software Engineering*, Wuhan, China, 11-13 December 2009.

[7] Zhang J, Su Y, Zhang M. “Exposing digital video forgery by ghost shadow artifact”. *Proceedings of the 1th ACM workshop on multimedia in forensics Conference*, Beijing, China, 23 October 2009.

[8] Hu Y, Li CT, Wang Y, Liu BB. “An improved fingerprinting algorithm for detection of video frame duplication forgery”. *International Journal of Digital Crime and Forensics*, 4(3), 64-76, 2013.

[9] Lin GS, Chang JF, Chuang CH. “Detecting frame duplication based on spatial and temporal analysis”. *6th International Conference on Computer Science & Education Conference*, Singapore, Singapore, 3-5 August 2011.

[10] Sun T, Wang W, Jiang X. “Exposing video forgeries by detecting double MPEG compression”. *IEEE International Conference on Acoustics, Speech and Signal Processing*, Kyoto, Japan, 25-30 March 2012.

[11] Subramanyam AV, Emmanuel S. “Video forgery detection using HOG features and compression properties”. *IEEE 14th International Workshop on Multimedia Signal Processing (MMSP)*, Banff, AB, Canada, 17-19 September 2012.

[12] Chao J, Jiang X, Sun T. “A novel video inter-frame forgery model detection scheme based on optical flow consistency”. *The International Workshop on Digital Forensics and Watermarking Conference,* Shanghai, China, 31 October 2012.

[13] Lin GS, Chang JF. “Detection of frame duplication forgery in videos based on spatial and temporal analysis”. *International Journal of Pattern Recognition and Artificial Intelligence,* 26 (7), 1-18, 2013.

[14] Liao SY, Huang TQ. “Video copy move forgery detection and localization based on Tamura texture features”. *6th International Congress on Image and Signal Processing,* Hangzhou, China, 16-18 December 2013.

[15] Lin CS, Tsay JJ. “A passive approach for effective detection and localization of region-level video forgery with spatio-temporal coherence analysis”. *Digital Investigation,* 11(2), 120-140, 2014.

[16] Su L, Huang T, Yang J. “A video forgery detection algorithm based on compressive sensing”. *Multimedia Tools and Applications*, 74(17), 6641-6656, 2015.

[17] Yang J, Huang T, Su L. “Using similarity analysis to detect frame duplication forgery in videos”. *Multimedia Tools and Applications,* 75(4), 1793-1811, 2016.

[18] Yang X, Cheng KTT. “Local difference binary for ultrafast and distinctive feature extraction”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1), 188-194, 2014.

[19] Calonder M, Lepetit V, Strecha C, Fua P. “Brief: Binary Robust Independent Elementary Features”. *11th European Conference on Computer Vision*, Heraklion, Greece, 5-11 September 2010.

[20] Rublee E, Rabaud V, Konolige K, Bradski G. “ORB: an Efficient Alternative to SIFT or SURF”. *In Proceeding of ‘International Conference on Computer Vision (ICCV)’*. Barcelona, Spain, 6-13 November 2011.

[21] Leutengger S, Chli M, Siegwart RY. “BRISK: Binary Robust Invariant Scalable Keypoints”. *In Proceeding of ‘International Conference on Computer Vision (ICCV)’*, Barcelona, Spain, 6-13 November 2011.

[22] Alahi A, Ortiz R, Vandergheynst P. “FREAK: Fast Retinal Keypoint”. *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 16-21 June 2012.

[23] Singh G, Singh K. “Video frame and region duplication forgery detection based on correlation coefficient and coefficient of variation”. *Multimedia Tools and Applications,* 78(7), 11527-11562, 2019.

[24] Ulutas G, Ustubioglu B, Ulutas M, Nabiyev V. “Frame duplication/mirroring detection method with binary features”. *IET Image Processing*, 11(5), 333-342, 2017.

[25] Ulutas G, Ustubioglu, B, Ulutas M, Nabiyev V. “Frame duplication detection based on bow model”. *Multimedia Systems,* 24(5), 549-567, 2018.

[26] Yao Y, Shi Y, Weng S, Guan, B. “Deep learning for detection of object based forgery in advanced video”. *Symmetry,* 10(1), 1-10, 2018.

[27] Bakas J, Bashaboin A K, Naskar R. "MPEG Double Compression Based Intra-Frame Video Forgery Detection using CNN". *2018 International Conference on Information Technology (ICIT)*, Bhubaneswar, India, 10-12 January 2018.

[28] Long C, Basharat A, Hoogs A. “A coarse to fine deep convolutional neural network framework for frame duplication detection and localization in forged videos”. *CVPR Workshops*, Salt Lake City, Utah, 18-22 June 2018.

[29] Raveendra M, Nagireddy K. “DNN based moth search optimization for video forgery detection”. *International Journal of Engineering and Advanced Technology*, 9(1), 1190-1199, 2019.

[30] D’Avino D, Cozzolino D, Poggi G. “Autoencoder with recurrent neural networks for video forgery detection”. *Electronic Imaging*, 7(1), 92-99, 2017.