IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 13, 2020, accepted March 9, 2020, date of publication March 16, 2020, date of current version March 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2980951

Classification of Authentic and Tampered Video Using Motion Residual and Parasitic Layers

MUBBASHAR SADDIQUE[1], KHURSHID ASGHAR[1], USAMA IJAZ BAJWA[2], MUHAMMAD HUSSAIN[3], HATIM A. ABOALSAMH[3], AND ZULFIQAR HABIB[2]

[1]Department of Computer Science, University of Okara, Punjab 56300, Pakistan
[2]Department of Computer Science, COMSATS University Islamabad, Lahore Campus, Lahore 54700, Pakistan
[3]Department of Computer Science, King Saud University, Riyadh 11451, Saudi Arabia

Corresponding author: Khurshid Asghar (khasghar@uo.edu.pk)

This work was supported in part by the Higher Education Commission (HEC) Pakistan under International Research Support Initiative Program (IRSIP) under Grant # 1-8/HEC/HRD/2017/8030, for the visit at Korea University, South Korea, in part by the European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie Grant for PDE-GIR Project under Agreement # 778035, and in part by the King Saud University, Riyadh, Saudi Arabia, under Researchers Supporting Project # RSP-2019/109.

ABSTRACT These days, videos can be easily recorded, altered and shared on social and electronic media for deception and false propaganda. However, due to sophisticated nature of the content alteration tools, alterations remain inconspicuous to the naked eye and it is a challenging task to differentiate between authentic and tampered videos. During the process of video tampering the traces of objects, which are removed or modified, remain in the frames of a video. Based on this observation, in this study, a new method is introduced for discriminating authentic and tampered video clips. This method is based on deep model, which consists of three types of layers: motion residual (MR), convolutional neural network (CNN), and parasitic layers. The MR layer highlights the tampering traces by aggregation of frames. The CNN layers encode these tampering traces and are learned using transfer learning. Finally, parasitic layers classify the video clip (VC) as authentic or tampered. The parasitic layers are learned using an efficient learning method based on extreme learning theory; they enhance the performance in terms of efficiency and accuracy. Intensive experiments were performed on various benchmark datasets to validate the performance and the robustness of the method; it achieved 98.89% accuracy. Comparative analysis shows that the proposed method outperforms the state-of-the-art methods.

INDEX TERMS Spatial forgery detection, motion residual, deep learning, extreme learning machine, parasitic learning.

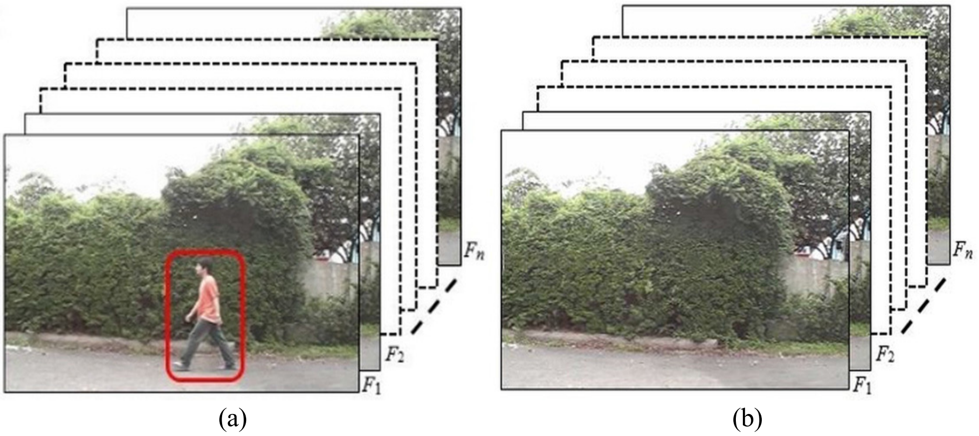
I. INTRODUCTION

In the digital era of 21st century, mobile phones, personal digital assistants (PDAs) and digital camcorders are easily available to acquire the videos. Moreover, these videos can be redistributed for different purposes like video conferences, surveillance systems, information propagation to the media houses and social media websites. The quality and contents of the videos can be modified with different video editing tools. With the influx of these user-friendly video editing tools, the novice user can alter the contents of the videos to make false propaganda on the social media for their own purposes [1], [2]. Now a day, tampering detection in videos is extremely difficult with naked eyes.

Videos tampering attacks can be categorized as: (i) spatial domain, (ii) temporal domain, and (iii) spatio-temporal domain [3]. In spatial domain, different objects can be added, removed or replaced within a video frame or a series of frames, whereas in the temporal domain, a number of frames are added, removed or replaced from the video [4]. Spatio-temporal domain is the combination of spatial and temporal domain.

Many techniques are proposed to detect forgery in images [5]–[7]. However, these cannot be considered effectively in the present form for the detection of forgery in the frames of videos due to the following reasons. Firstly, videos are encoded and compressed before storage and transmission

The associate editor coordinating the review of this manuscript and approving it for publication was Yudong Zhang.



(a) (b)

FIGURE 1. Tampering in spatial domain. (a) Authentic video frames. (b) Tampered video frames.

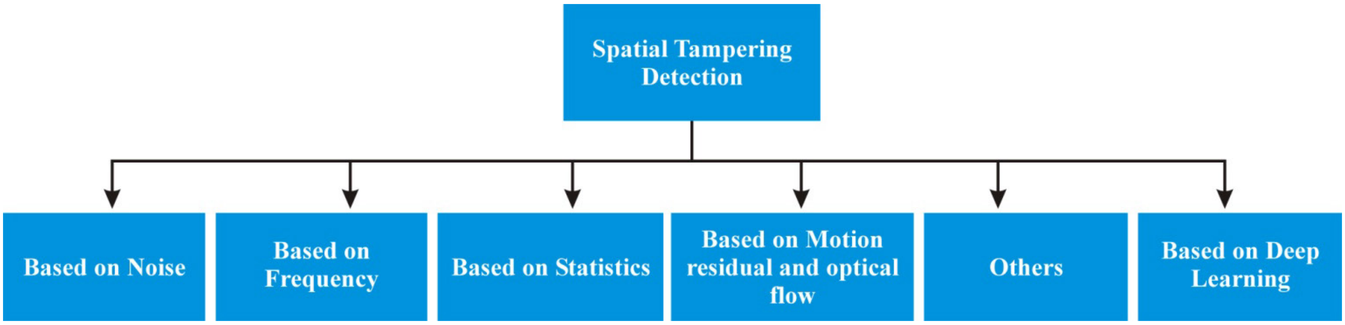


FIGURE 2. Categories of spatial tampering detection techniques.

to reduce the amount of data in video frames. Secondly, computational complexity of these techniques becomes very high when applied to a large number of video frames. Thirdly, forgery traces are also available in consecutive frames of a video. Such features are not available in case of images.

In spatial domain, tampering can be done in two different ways i.e., copy move and splicing. In copy-move the object/s is copied and pasted in frames of the same video and in case of removing the object/s from the video, the region/s is filled with the neighboring pixels from the same frame of that video whereas, in case of splicing the tampering is done with the object/s from different video/s. The focus of this paper is on tampering detection in the spatial domain of videos. An example of spatial tampering is shown in Fig. 1. Authentic video frames are presented in Fig. 1 (a) whereas Fig. 1 (b) describes frames of the tampered video. The actual information is concealed by removing the object in red rectangle shown in Fig 1(a) from all the frames F_1, F_2, …, F_n and filled that areas with a block of pixels from the same frame of a video. In this way, the viewers are misguided by false information. The purpose of this type of tampering is not simple retouching or format change, but to hide the facts for propaganda or criminal intentions, which is dangerous and has negative impact on society.

A. RELATED WORK

The existing technivided into two categories: Active and Passive [8]. Passive techniques do not need any pre-embedded data such as watermark and signatures unlike active techniques. The focus of this study is on passive techniques in spatial domain and as such we give an overview of the state-of-the-art related to this problem in the following paragraphs.

Different techniques proposed to detect tampering in spatial domain can be divided into various categories based on the types of features as shown in Fig. 2. In first category the features are extracted based on noise. Kobayashin *et al.* [9] employed noise characteristics, Hyun *et al.* [10] detected forgery using sensor patter noise (SPN). Panday *et al.* [11] worked with SIFT features, noise residual and correlation to detect copy-move forgery. Goodwin and Chetty [12] also used noise residual, quantization features and their transformation in cross-model subspace to detect the copy-move forgery.

In second category the features are extracted based on frequency domain. Hsu *et al.* [13] used wavelet co-efficient thresholding and Bayesian Classifier. Su *et al.* [14] applied exponential-Fourier moments (EFM) for localizing the duplicating region in the frames of the video. In [15] moment feature of wavelet co-efficient and optical flow are combined with SVM to detect the facial expression re-enacted forgery (FERF).

The features are calculated thourgh statistical method in third category. Richao *et al.* [8] worked on statistical features. Bagiwa *et al.* [16] explored statistical correlation of blurring artifact. Singh and Aggarwal [17] used pixels correlation, noise inconsistency and discrete fractional fourier transformation. Singh and Singh [18] exploited the correlation coefficient to find the duplicated regions in the videos.

Fourth category of techniques is worked on optical flow and motion residual. Bidokhti and Ghaemmaghami [19] proposed a technique based on optical flow to detect a copy-move forgery from MPEG videos In [20] block based motion estimation is used to extract motion from the adjacent frames and then the magnitude and orientation is employed to differentiate the authentic and forged video. Al-sanjary *et al.* [21] proposed optical flow inconsistencies and dynamic time warping (DTW) matching algorithm to detect copy-move forgery in videos. Bestagini *et al.* [22] utilized Zero-motion video residual. Chen *et al.* [23] used motion residual and steganography features to detect the video forgery.

In fifth category the Subramanyam *et al.* [24] exploited histogram of oriented gradients (HOG) features. The HOG and its variants employ gradient orientation, which cannot describe the local texture micro-patterns and variations effectively. Also, it is not robust against noise [25] and does not take into account the strength of edges. Su and Li [26] detected copy-move forgery in first frame of the video by employing MISIFT and used spatio-temporal context learning to detect the forged areas from remaining frames of the video. Su *et al.* [27] used K-SVD (k-singular value decomposition) algorithm and K-means. The technique presented in [28] exposed the forgery in videos that have ballistic motion.

All above discussed techniques although achieved good accuracy but with some limitations such as, these techniques only work on specific formats, specific resolutions, selected datasets and handcrafted features are used to detect the forgery. Similarly Yao *et al.* [29] utilized a CNN to extract high dimension features and used absolute difference between successive frames to cut down the temporal redundancy. A max pooling and high pass filter layers are used to minimize the computational complexity and to increase the residual, which left during the tampering process respectively. Zampoglou *et al.* [30] employed Q4 and Cobalt forensic filters with pre-trained GooLeNet and ResNet networks to detect the video forgery. These techniques although produced good results, however cannot work well in presence of small size tampering.

B. MULTIMEDIA FORENSIC USING DEEP LEARNIN

In recent years, deep neural networks, such as deep belief network (DBN) [31], stacked auto encoder (SE) [32] and convolutional neural network (CNN) [33]–[35] have shown the capability of learning robust feature representations. This allows to generalize across a wide variety of computer vision (CV) tasks such as image classification [36], speech recognition [37], image forensic [6] etc. However, a deep model requires a lot of time, a large amount of data and a powerful computing environment for its training from scratch. The collection of a large amount of domain specific data and its labeling is a tedious and costly task, and even acquiring a sufficient amount of data may not be practical in many cases [38], [39]. In such cases, the only way to employ deep learning is to use transfer learning (TL), which unlocks a new stream of techniques. In TL, a deep model is trained using a dataset from a related domain and then it is employed for the application under consideration. There are two main approaches for TL: (1) first pre-train a deep model using the data from a related domain and then fine-tune the weights using the data from the domain of the problem, (2) use trained deep model as a feature extractor by freezing all layers other than the last classification layers, extract features and pass them to a classifier such as support vector machine (SVM) for classification [40]. The second approach has been effectively employed for many recognition and classification tasks [41]. For video tamper detection problem, enough data is not available to train a deep CNN model from scratch. As such, the second TL approach is employed to propose a method based on deep model, which comprises three types of layers i.e., motion residual, CNN and parasitic layers. This deep model is used in our method for the classification of authentic and tampered video clips (VCs).

The outstanding performance of deep CNN in many applications motivated the research in this direction and many deep CNN models such as AlexNet [36], GoogleNet [42] and VGG-16 [35] have been proposed. These models have shown far better performance than hand-engineered techniques in many applications. Different existing deep CNN models are examined and selected VGG-16 for our deep model because its convolutional layers contain kernels of small size i.e., 3×3, which is suitable to characterize the small tampering traces. We employed VGG-16 model, pre-trained on the large scale ImageNet datase; by removing its last fully connected (FC) layers, we used its leftover layers as CNN layers in our deep model. For decision making, the parasitic layers in the deep model rely on the discriminant features extracted by the CNN layers and motion residual layer, just like parasites in biology [43], therefore we call them as parasitic layers. The parasitic layers involve a small number of learnable weights, which can be easily learned using the available data for video tamper detection.

This study consists of following contributions: (i) For video tamper detection, an efficient and robust method has been introduced, (ii) for this method, a deep model has been proposed; it consists of three types of layers: motion residual layer, CNN layers, and parasitic layers, (iii) an efficient training method has been introduced for parasitic layers, which is based on extreme learning theory and improves the overall performance in terms of accuracy and efficiency. The proposed method based on the deep model gives better accuracy (98.89%) and efficiency for the classification of authentic and tampered VCs on different datasets than the state-of-the-art methods. According to our knowledge, this type of efficient and robust method based on deep learning has been introduced first time for video tamper detection

C. MOTIVATION AND GOALS

When a forger tampers video by adding or removing any object, the traces/shapes of that object remain in the video, although it is post-processed to conceal the tampering traces. These shapes/patterns can be learned through deep networks from the huge amount of data, but it is a time-consuming task and difficult to label all the data. Furthermore, the learning of the huge number of weights at each layer is required, which need more time. Similarly, another limitation is that a large number of tampered videos are not available to learn all these shapes. Based on these limitations, transfer learning is explored.

The objective of this study is to investigate the capability of deep CNN for classification of authentic and tampered videos and target to answer these questions: (a) Which model of deep learning should be employed? (b) Which part of the model should be used for feature extraction and representation of the tampered traces? (c) How many layers should be transferred in order to obtain best performance? (d) Which classifier is best to classify authentic and tampered videos? (e) How to modify the existing deep CNN model to make a new model for classification of authentic and tampered videos?

D. ORGANIZATION OF STUDY

The rest of the study is organized as follows. In section 2, the methodology is described in detail. Evaluation detail and datasets are elaborated in section 3. Extensive experiments detail and their analysis are presented in section 4 and 5. Finally, section 6 concludes this study with future work.

II. PROPOSED METHOD

In this section, a method based on deep learning is presented for the classification of VCs as authentic or tampered. The method takes a VC as an input and gives the decision whether it is authentic or not. A video is segmented into non-overlapping VCs, and its authenticity is validated based on whether all VCs are authentic. If a video is tampered, then this approach not only detects tempering but also localizes the probable tempered frames.

A video (V) containing N frame is divided into non-overlapping VCs, each consisting of W frames, where $W = 2m + 1$ for some integer m. We tested W with $m = 4$, 6, 8, 10, 12, 14. The $W = 9$ with $m = 4$ was selected as the initial size of VCs for experiments because in order to create plausible tampering "at least" 10 frames are manipulated (because manipulation of fewer frames will not accomplish anything meaningful) [44]. Empirically, we found that $W = 21$ with $m = 10$ results in better detection performance; it means that more than 10 frames are necessary for reliable detection; the frames in a VC in addition to tampered frames from the context and this contextual information lead to better detection performance. In case, the total number of frames of a video is not exact multiple of 21, the last VC consists of less than 21 frames. In this case, the last VC is created by taking the last 21 frames of the video. In this way, the method authenticates each VC one by one to verify complete video.

The method consists of a deep CNN model, which comprises three types of layers: (i) motion residual (MR) layer, (ii) CNN layers which involve convolutional, pooling and fully connected layers, (iii) parasitic layers. The overall architecture is shown in Fig. 3. The MR layer takes a VC of size W as input and calculates the motion residual for each of R, G and B channels, and concatenates them as a 3- channel activation, which is passed to CNN layers. The MR highlights the tampering traces by aggregation of frames. The CNN layers compute the hierarchical representation of the VC. Finally, parasitic layers work as a classifier and predict whether the VC is authentic or tampered. The detail of each layer is elaborated in the following subsections.

A. MOTION RESIDUAL (MR) LAYER

The videos can be tampered by adding and/or deleting the objects with sophisticated tools, but to hide the tampering traces (lines, edges), different operations such as video in-painting, contrast adjustment, blurring, and video layer fusion are performed. In this way, some specific statistical properties of the tampered videos are altered. These alterations can help in tampering detection, if they are modelled well. One way to model these alterations is to use MR of the video sequences [23], which is calculated by MR layer. The visual description of MR layer is shown in Fig. 4. The calculation of motion residual is elaborated in algorithm 1.

Algorithm 1 The Computation of Motion Residual (MR)

```
Input: Non-overlapping video clip (VC) of size $W$
Output: Motion residual of VC
Procedure:
    1. Identify the central frame $F_c$ of VC
    2. Apply aggregated function $\Im$ on VC and compute
       aggregated frame AF
    3. Compute motion residual MR such that $MR = |F_c - AF|$
```

An aggregated operation is performed over a VC to get an aggregated frame (AF) which is defined as

$$AF(x, y) = \Im[F_{c-NB}(x, y), \ldots, F_c(x, y), \ldots, F_{c+NB}(x, y)], \tag{1}$$

where F_c is the central frame, $NB = m$. The operator $\Im$ is an aggregated function that processes the pixels in the same positions of all frames in the VC.

The MR of VC is calculated as follows:

$$MR(x, y) = |F_c(x, y) - AF(x, y)|, \tag{2}$$

where $|.|$ denotes the absolute value. In this study, four aggregated functions $\Im_{Minimum}$, $\Im_{Maximum}$, $\Im_{Median}$, and $\Im_{Average}$ are

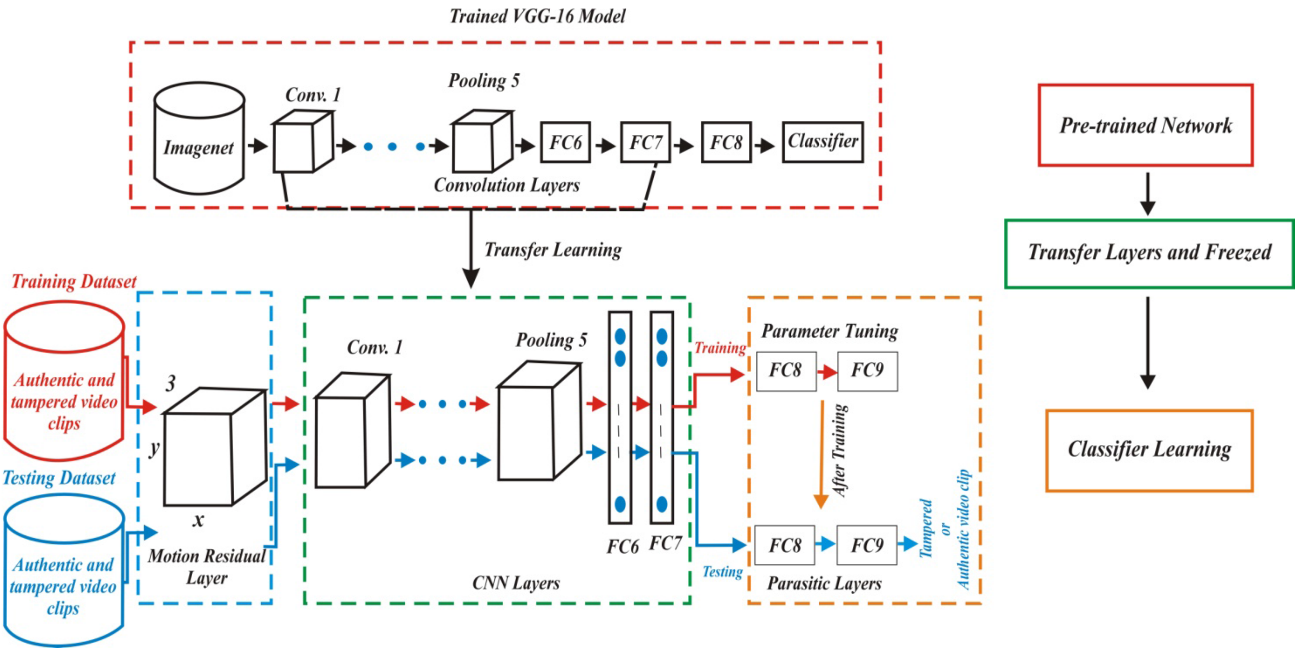


FIGURE 3. Architecture of the proposed method for the classification of authentic and tampered VC.

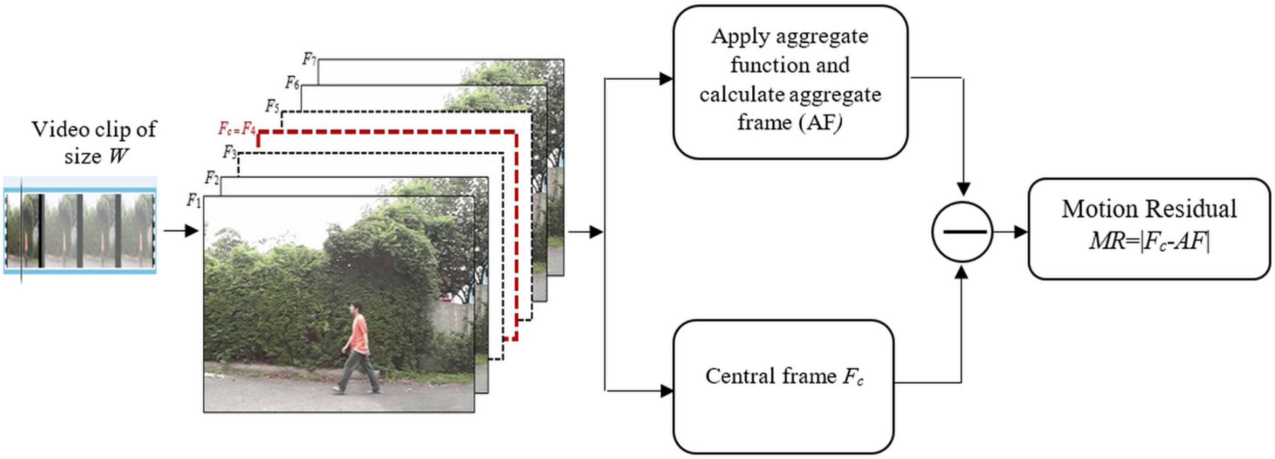


FIGURE 4. Motion Residual (MR) layer of VC of size $W = 7$. Each frame has three channels R, G and B. MR is calculated for each channel and as such MR has three channels.

examined. These functions are defined as:

$$\Im_{Minimum}(x,y) = Min\,(F_{c-NB}(x,y),\ldots,F_c(x,y),\ldots,F_{c+NB}(x,y)), \tag{3}$$

$$\Im_{Maximum}(x,y) = Max\,(F_{c-NB}(x,y),\ldots,F_c(x,y),\ldots,F_{c+NB}(x,y)), \tag{4}$$

$$\Im_{Median}(x,y) = Median\,(F_{c-NB}(x,y),\ldots,F_c(x,y),\ldots,F_{c+NB}(x,y)), \tag{5}$$

$$\Im_{Average}(x,y) = \left(\sum_{i=1}^{w} F_i(x,y)\right)/w, \tag{6}$$

where (x, y) is the position of a pixel in all frames of VC. Please note that according to equations (3) to (6) the $AF(x, y) \in \{0, \ldots, 255\}$ which yields MR, where $MR(x, y) \in \{0, \ldots, 255\}$ There are four scenarios for recording a video:

- Scene and camera both are static
- Scene is static but camera is moving
- Scene is moving and camera is static
- Scene and camera both are moving

In the first scenario, a small change occurs in the frames of a VC due to acquisition noise. The distribution of this noise is symmetric in all the frames of VC [10], [23]. Therefore, MR is almost zero in case of authentic VC because $F_{c-NB}(x, y) \cong \cdots \cong F_c(x, y) \cong \cdots \cong F_{c+NB}(x, y)$. However, in a tampered VC, the tampered regions in all the frames of the VC are not consistent and due to these inconsistent tampered regions, the noise patterns are not same in the frames of VC. Similarly, in other three scenarios the noise patterns of the forged VCs are different from those of authentic VCs. As such, MR is helpful to highlight these changes which are introduced during the tampering process. With the increase/decrease of motion in a scene, though there is a variation in the patterns of MRs of tampered VCs, they are different from those of authentic VCs. In all The MR layer takes VC of any resolution and gives the output rescaled to 224 × 224 × 3. Then this activation of MR is fed forward to CNN layers for further processing.

B. CNN LAYERS

A CNN [43] is a type of deep learning model which has shown excellent performance on pattern recognition tasks such as hand-written digit classification, image classification and human action recognition [36], [40]. It is a hierarchical learning model with multiple hidden layers, which transforms the input volume to output categories. Its architecture consists of three main types of layers: convolutional layer, pooling layer, and fully-connected layer. One limitation of deep model is over-fitting when it is learned with small datasets. However, the over-fitting can be avoided by increasing the size of the training data, but it is a difficult and expensive task to arrange a large amount of annotated data. In this situation, transfer learning comes into play and solves this issue by using pre-trained model to generate a new architecture [45]. In this study, using pre-trained VGG-16 model, a method based on deep model, is proposed for the classification of authentic and tampered VCs. In the proposed method, the input layer and last two layers (FC8, softmax) of VGG-16 model are replaced. The input layer is replaced with MR layer which has been discussed earlier. The last two higher layers are replaced with parasitic layers which are discussed later in section parasitic layers. The intermediate layers of VGG-16 model, other than the first and the last two layers are freezed (see Table 1) and used as CNN layers in the proposed method.

The CNN layers are trained on an ILSVR dataset [46] by using stochastic gradient descent (SGD) [47]. The detail of key parameters of these layers is shown in Table 1. In this table, four kind of components are described in terms of kernel dimension, number of kernels, stride and padding. The Conv, pooling and FC represents the convolution, pooling and fully connected layers respectively. Furthermore, each convolution layer has a rectified linear unit as the activation function. The deep and discriminant representation of the tampered traces is obtained from the fully connected layer FC7. Then the activation of FC7 layer is fed forward to parasitic layers.

C. PARASITIC LAYERS

Detection of video tampering is a two class problem (authentic or tampered). The last fully connected layers of VGG-16 learn the task specific knowledge and are trained using SGD algorithm, which needs a huge volume of data and a lot of time. As an attempt to address this issue, the FC8 and softmax layers of the VGG-16 are replaced with two new layers, which depend on the activation of FC7 layer and thus the name parasitic layers. These layers are learned using an extreme learning machine (ELM) [48], [49] algorithms, which improve the efficiency and accuracy of classification.

TABLE 1. The parameters detail of CNN layers.

Layer No.	Layer Name	Kernel Dimension	No. of Kernels	Stride	Padding
1.	Conv. 1_1	3×3×3	64	1	1
2.	Conv. 1_2	3×3×64	64	1	1
3.	Pooling1			2	0
4.	Conv. 2_1	3×3×64	128	1	1
5.	Conv. 2_2	3×3×128	128	1	1
6.	Pooling2			2	0
7.	Conv. 3_1	3×3×128	256	1	1
8.	Conv. 3_2	3×3×256	256	1	1
9.	Conv. 3_3	3×3×256	256	1	1
10.	Pooling3			2	0
11.	Conv. 4_1	3×3×256	512	1	1
12.	Conv. 4_2	3×3×512	512	1	1
13.	Conv. 4_3	3×3×512	512	1	1
14.	Pooling4			2	0
15.	Conv. 5_1	3×3×512	512	1	1
16.	Conv. 5_2	3×3×512	512	1	1
17.	Conv. 5_3	3×3×512	512	1	1
18.	Pooling5			2	0
19.	FC6	7×7×512	4096	1	0
20.	FC7	1×1×4096	4096	1	0

As there are two classes in the problem of video tampering detection, therefore, FC9 contains only two neurons. The number L of neurons in FC8 depends on the learning choice such as learning algorithm 1 or learning algorithm 2. In the case of learning algorithm 1, the neurons of FC8 use ordinary activation functions such as sigmoid, hard-limit, sine functions and their number are fixed using cross validation. When the learning algorithm 2 is used, the neurons of FC8 use a kernel function such as RBF, linear, or polynomial kernel and their number are fixed by the number of training examples. The fully connected layer FC7 acts as an input layer for both options. These learning algorithms save the time and reduce the computational cost significantly because they learn the parameters of FC8 and FC9 by solving the closed form optimization problem.

In the proposed method, the learnable layers are CNN layers and parasitic layers. The CNN layers are pre-trained using ImageNet dataset and the learnable part consists of only parasitic layers, which involve a small number of parameters and the available data is enough to learn these layers, so there is no problem of over-fitting.

1) LEARNING ALGORITHM 1

This learning algorithm is motivated by ELM [49]. In this algorithm, during the learning process, the weights and biases of FC8 layer are randomly assigned and weights of FC9 are computed by solving a closed form problem. Let the weights and bias of the 'i^{th}' neuron of FC8 be denoted by the row vector $w_i^8 = \left[w_{i1}^8 \; w_{i2}^8 \; \ldots \; w_{is}^8 \right]$ and b_i^8 as shown in Fig. 5. Also, let the output of the layer FC7 be denoted by s-dimensional vector X^7, then the output L-dimension vector Z of FC8 is

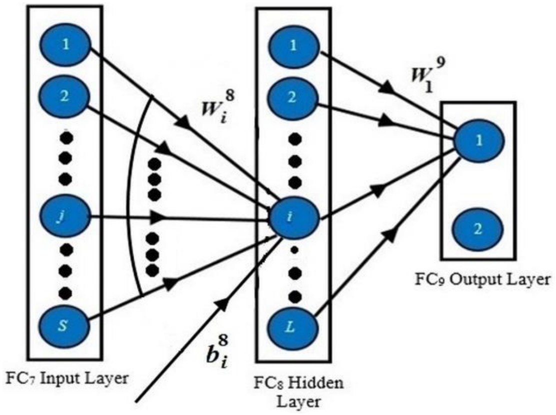


FIGURE 5. The architecture of parasitic layers.

calculated as:

$$Z = f(\mathbf{W^8}X^7 + b),\ \text{where } \mathbf{W^8} \text{ and } b \text{ is weight matrix and bias vector of FC8 such that :} \quad (7)$$

$$\mathbf{W^8} = \left[\, (w_1^8)^T \ (w_2^8)^T \ \ldots \ (w_L^8)^T \,\right]^T$$

$$\text{and } b = \left[\, b_1 \ b_2 \ \ldots \ b_L \,\right]^T, \quad (8)$$

T and f are the transpose and the activation function (sigmoid, sine and hard-limit) respectively, which are defined by equations (13), (14) and (15). Let $\mathbf{W^9}$ be the weights of FC9 then the output of FC9 is a 2-dimension vector such that $\bar{Y} = \mathbf{W}^9 Z$. If $X_1^0, X_2^0, \ldots, X_n^0$ are the training examples, then the corresponding outputs of FC8 are $Z_1, Z_2, \ldots, Z_n$, which are combined into a $L \times n$ matrix $\mathbf{Z}$ such that $\mathbf{Z} = [Z_1, Z_2, \ldots, Z_n]$. Therefore, the output $2 \times n$ matrix of FC9 is:

$$\bar{\mathbf{Y}} = \mathbf{W}^9\mathbf{Z} \quad (9)$$

If $y_1, y_2, \ldots, y_n$ are the labels of training examples $X_1^0, X_2^0, \ldots, X_n^0$ then $\mathbf{Y} = [y_1, y_2, \ldots, y_n]$ is a $2 \times n$ matrix of the original labels. We expect that the predicted labels must match with $\mathbf{Y}$ i.e.:

$$\mathbf{Y} = \bar{\mathbf{Y}} \quad (10)$$

From equation (9) and (10), we get,

$$\mathbf{Y} = \mathbf{W^9Z} \quad (11)$$

In equation (11), $\mathbf{W^9}$ is unknown and is calculated by solving the equation (11) as follows:

$$\mathbf{W}^9 = \mathbf{YZ}^{\dagger}, \quad (12)$$

where $\mathbf{Z}^{\dagger} = \mathbf{Z}^T(\frac{1}{c} + \mathbf{ZZ}^T)^{-1}$ is the Moore-Penrose generalized inverse of $\mathbf{Z}$ and for stable solution a regularized term $1/C$ is added. The learning algorithm 1 is outlined in Algorithm 2. The equations of the activation functions (sigmoid, hard-limit and sine) are as follows.

Sigmoid function :

$$G(a, b, X) = \frac{1}{1 + e^{-(a.X+b)}}, \quad (13)$$

Hard − limit function :

$$G(a, b, X) = \begin{cases} 1, & \text{if } a.X - b > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Sine function :

$$G(a, b, X) = \sin(a.X + b). \quad (15)$$

Algorithm 2 Learning of Parasitic Layers Using Simple ELM

```
Input: Activations of FC7 layer $X_1^7, X_2^7, \ldots, X_n^7$, activation
       function $f$, number $L$ of hidden neurons in FC8 layer
Output: Parameters of the parasitic layers
Procedure:
    1. Assign random weights $W^8$ and biases $b^8$ to the $L$ hidden neurons of FC8
    2. Calculate the activations $Z_1, Z_2, \ldots, Z_n$ of $X_1^7, X_2^7, \ldots, X_n^7$ using equation (7) and form matrix $\mathbf{Z} = [Z_1, Z_2, \ldots, Z_n]$
    3. Compute $\mathbf{Z}^{\dagger}$, the Moore-Penrose generalized inverse of $\mathbf{Z}$
    4. Compute $\mathbf{W^9}$ using equation (12)
    5. Return $\mathbf{W^8}$, $b^8$, and $\mathbf{W^9}$
```

2) LEARNING ALGORITHM 2

This learning algorithm is motivated by kernel ELM [48]. In this learning algorithm, the number L of neurons in FC8 layer is equal to the number n of training examples i.e., $L = n$. The weights and bias of the 'i^{th}' neuron at FC8 are assigned such that $W_i^8 = X_i^7$ and $b_i = 0$ where X_i^7 is the output of the training example X_i through FC7. The output matrix $\mathbf{Z}$ of FC8 layer is calculated using a kernel function such as Radial Bases Function (RBF), linear and polynomial functions. If X is any input pattern, and X^7 is the output through FC7, then the corresponding output Z from FC8 is:

$$Z(X^7) = \begin{bmatrix} K(X^7, X_1^7) \\ K(X^7, X_2^7) \\ \cdot \\ \cdot \\ \cdot \\ K(X^7, X_n^7) \end{bmatrix}. \quad (16)$$

Let $\mathbf{W^9}$ be the weights of FC9 then the output of FC9 is a 2-dimension vector such that $\tilde{Y} = \mathbf{W}^9 Z$. If $X_1^0, X_2^0, \ldots, X_n^0$ are the training examples and their activations by FC7 layer are $X_1^7, X_2^7, \ldots, X_n^7$ then their activations by FC8 are $Z_1, Z_2, \ldots, Z_n$, which are arranged into $n \times n$ symmetric kernel matrix $\mathbf{Z} = [Z_1, Z_2, \ldots, Z_n]$ i.e.,:

$$\mathbf{Z} = \begin{bmatrix} K(x_1^7, x_1^7) & K(x_2^7, x_1^7) & \bullet & \bullet & \bullet & K(x_n^7, x_1^7) \\ K(x_1^7, x_1^7) & K(x_2^7, x_1^7) & \bullet & \bullet & \bullet & K(x_n^7, x_1^7) \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ K(x_1^7, x_n^7) & K(x_2^7, x_n^7) & \bullet & \bullet & \bullet & K(x_n^7, x_n^7) \end{bmatrix}. \quad (17)$$

Then, the output matrix of FC9 is:

$$\widetilde{\mathbf{Y}} = \mathbf{W^9 Z} \tag{18}$$

If $y_1, y_2, \ldots, y_n$ are the labels of training examples $X_1^0, X_2^0, \ldots, X_n^0$ then $\mathbf{Y} = [y_1, y_2, \ldots, y_n]$ is a $2 \times n$ dimension matrix of original labels. We expect that the predicted labels must match with $\mathbf{Y}$ i.e.,:

$$\mathbf{Y} = \widetilde{\mathbf{Y}} \tag{19}$$

Equations (18) and (19) can be combined as:

$$\mathbf{Y} = \mathbf{W^9 Z} \tag{20}$$

In equation (20), $\mathbf{W^9}$ is unknown and can be calculated by solving the equation (20) because $\mathbf{Z}$ is a symmetrical matrix such that:

$$\mathbf{W^9} = \mathbf{YZ}^{-1} \tag{21}$$

For stable solution a regularized term $1/C$ is added in equation (21) as:

$$\mathbf{W^9} = \mathbf{Y}(\frac{1}{c} + \mathbf{Z}^{-1}) \tag{22}$$

The learning algorithm 2 is presented in Algorithm 3. The equations of the kernel functions RBF, linear, and polynomial are as follows:

$$\text{RBF Kernel}: K(X, X') = \exp\left(-\frac{\|X - X'\|^2}{2\sigma^2}\right), \tag{23}$$

$$\text{Linear Kernel}: K(X, X') = X.X', \tag{24}$$

$$\text{Polynomial Kernel}: K(X, X') = (X.X' + b)^d, \tag{25}$$

Algorithm 3 Learning of Parasitic Layers With Kernel ELM

```
Input: Activations of FC7 layer $X_1^7, X_2^7, \ldots, X_n^7$, Kernel function
Output: Parameters of the parasitic learning layer
Procedure:
    1. Assign the value of neurons such that $W_i^8 = X_i^7$ and $b_i^8 = 0$ where $i = 1, 2, \ldots, n$
       $\mathbf{w^8} = [w_1^8, w_2^8, \ldots, w_n^8]$, $\mathbf{b^8} = [b_1^8, b_2^8, \ldots, b_n^8]$
    2. Calculate the activations $Z_1, Z_2, \ldots, Z_n$ of $X_1^7, X_2^7, \ldots, X_n^7$ using equation (16) and form matrix $\mathbf{Z} = [Z_1, Z_2, \ldots, Z_n]$
    3. Calculate the output weight $\mathbf{W^9}$ using equation (22)
    4. Return $\mathbf{W}^8$, $b^8 = 0$ and $\mathbf{W^9}$
```

III. EVALUATION FRAMEWORK

This section first gives an overview of the datasets used for evaluation and then presents the detail of the evaluation protocol that are used for the validation of the proposed method.

A. DESCRIPTION OF DATASETS

The proposed method is validated on six (6) different datasets. The videos of these datasets have different resolutions including 320 × 240, 720 × 480, 768 × 576 and 720 × 1280. The formats of the videos are AVI, MOV, MP4 and WMV. Some datasets were captured with static cameras, some with moving cameras and some with both. The detail of these datasets is presented in Table 2. All the datasets containing tampered videos were created for research on video tamper detection and keeping in view this purpose, the videos were tampered to simulate the intentional video tampering scenarios.

The dataset D1 was captured with three types of cameras i.e., Cannon, Fuji and Nikon and it contains 100 authentic videos having frame rate of 30fps and resolution 320 × 240. It has AVI and WMV formats. The length of the videos varies between 4 to 15 seconds. The D2 dataset contains a total of 20 videos, in which 10 videos are authentic and 10 are forged. The frame rate is 30fps and length varies between 7 to 19 seconds. The dataset D3 has 14 authentic and 6 forged videos. The length varies from 3 to 17 seconds. The D4 dataset consists of 6 authentic and 121 forged videos. Five videos were captured with a static camera and one with moving camera. This dataset has frame rate of 30 and the length of the videos varies from 2 to 16 seconds. Each authentic video is forged with different geometric transformations (flipping, rotation, scaling and shearing) and post-processing operations (luminance, RGB, None). In this dataset, the forgery was done such that it could be visible and invisible with naked eyes.

The D5 dataset comprises of 40 videos, in which 20 videos are authentic and 20 are forged. The frame rate of this dataset is 29fps and the duration of videos is between 14 to 15 seconds. Static and moving cameras are involved to capture the videos. In this dataset the tampered videos are created with copy-move and splicing forgery. The dataset D6 was created from the datasets D1, D2, D3 and D5 such that authentic videos are taken from D1 and forged videos from datasets D2, D3 and D5. The D7 dataset consists of a total 150 authentic and 157 forged videos taken from the datasets D1 to D5. Authentic videos are collected from D1, D2, D3, D4 and D5 whereas forged videos are taken from D2, D3, D4 and D5.

B. EVALUATION PROTOCOL

The generalization and robustness of the method on different datasets D_i where $i = 2, 3, 4, 5, 6, 7$ are evaluated using 10-fold cross validation i.e., VCs from videos in each dataset are partitioned into 10 folds. Each fold is held out in turn for testing and remaining folds are used for training. The results are reported as the average and standard deviation of 10-folds. It is employed to ensure that the proposed method is not suffering from over-fitting.

Performance of our proposed method is evaluated using different evaluation measures such that true positive rate (TPR), true negative rate (TNR) and accuracy (AC). The t-SNE method [53] is introduced to give the discriminant

TABLE 2. Detail of all datasets.

Datasets	Types of forgery	Authentic	Forged	Frame Rate	Static/Moving Camera	Resolution	Format				Camera			Length (Sec.)
							AVI	MOV	MP4	WMV	Cannon	Fuji	Nikon	
D1 [49]	-	100	-	30	Static & Moving	320×240	60	40	-	-	41	30	29	4-15
D2 [20]	Copy-move	10	10	30	Static	320×240	20	-	-	-	-	-	-	7-19
D3 [11]	Splicing	14	6	30	Moving	720×480	15	-	-	5	-	-	-	3-17
D4 [50]	Copy-move & Splicing	6	121	25-30	Static & Moving	768×576	101	-	36	-	-	-	-	2-16
D5 [51]	Copy-move & Splicing	20	20	29	Static & Moving	720×1280	-	-	40	-	-	-	-	14-15
D6	Variable	100 (D1)	36 (D2+D3+ D5)	29-30	Static & Moving	Variable	71	40	20	5	41	30	29	2-19
D7	Variable	150 (D1+D2 + D3+D4+ D5)	157 (D2+D3+ D4+D5)	25-30		Variable	196	40	76	5	41	30	29	2-19

analysis of features used in proposed method. Assuming that the videos which are authentic are called negative cases and the video sequences have some tampered object(s) known as positive cases.

There are four possible categories to judge the binary (positive or negative) classification problem. (1) True positive (TP): positive samples that are classified as positive; (2) true negative (TN): negative samples that are classified as negative; (3) false positive (FP): negative samples that are classified as positive and, (4) false negative (FN): positive samples that are classified as negative. These metrics are defined as:

$$AC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{26}$$

$$TPR = \frac{TP}{TP + FN}, \tag{27}$$

$$TNR = \frac{TN}{TN + FP}, \tag{28}$$

IV. EXPERIMENTAL SETUP, RESULTS, COMPARISONS AND DISCUSSION

Several VCs are used to demonstrate the effectiveness and robustness of the proposed method. The experiments are performed with MATLAB 2016b. The hardware specification is as follows. CPU: Intel Xeon 2.67 GHz, Memory size: 16 GB, OS: Microsoft Windows 8.0.

A. HYPER PARAMETERS TUNING

Different hyper parameters like aggregative functions, VC sizes, classifiers, number of CNN layers, activations, and kernel functions are required to be tuned in the proposed method which affects the accuracy. The effect of these parameters is elaborated below in detail.

1) EFFECTIVENESS OF DIFFERENT AGGREGATIVE FUNCTIONS WITH DIFFERENT VIDEO CLIP (VC) SIZES W ON ACCURACY

The influence of VC size of W frames with different aggregated functions on accuracy is shown in Fig. 6. The activation of FC7 layer of CNN layers are used to select the best clip size W with different aggregated functions. The best accuracy is achieved 98.89% by $\Im_{Median}$ on VC of size $W = 21$, $NB = 10$, and $m = 10$. The accuracy using $\Im_{Median}$ is increasing steadily with the increase of clip size W. After clip size of $W = 21$, the accuracy starts decreasing, however, all other aggregated functions like maximum, minimum and average have lower accuracy as compared to the median function. Aggregated function $\Im_{Median}$ and $W = 21$ are used for all other experiments.

2) EFFECT OF DIFFERENT NUMBER OF LAYERS OF VGG-16 ON ACCURACY WITH DIFFERENT CLASSIFIERS

In this research, one of our objective was to find out how many layers of the VGG-16 model are required to freeze for transfer learning to achieve best accuracy for classification against authentic and tampered VCs. Activations of different layers of VGG-16 model are used with different classifiers. The best result is achieved on activation of FC7 layer with our proposed parasitic layers as shown in Fig. 7. The accuracy is decreased for all other classifiers, as the layers are increased. Based on experiments, the activation of FC7 layer is selected as input to proposed parasitic layers. Furthermore,

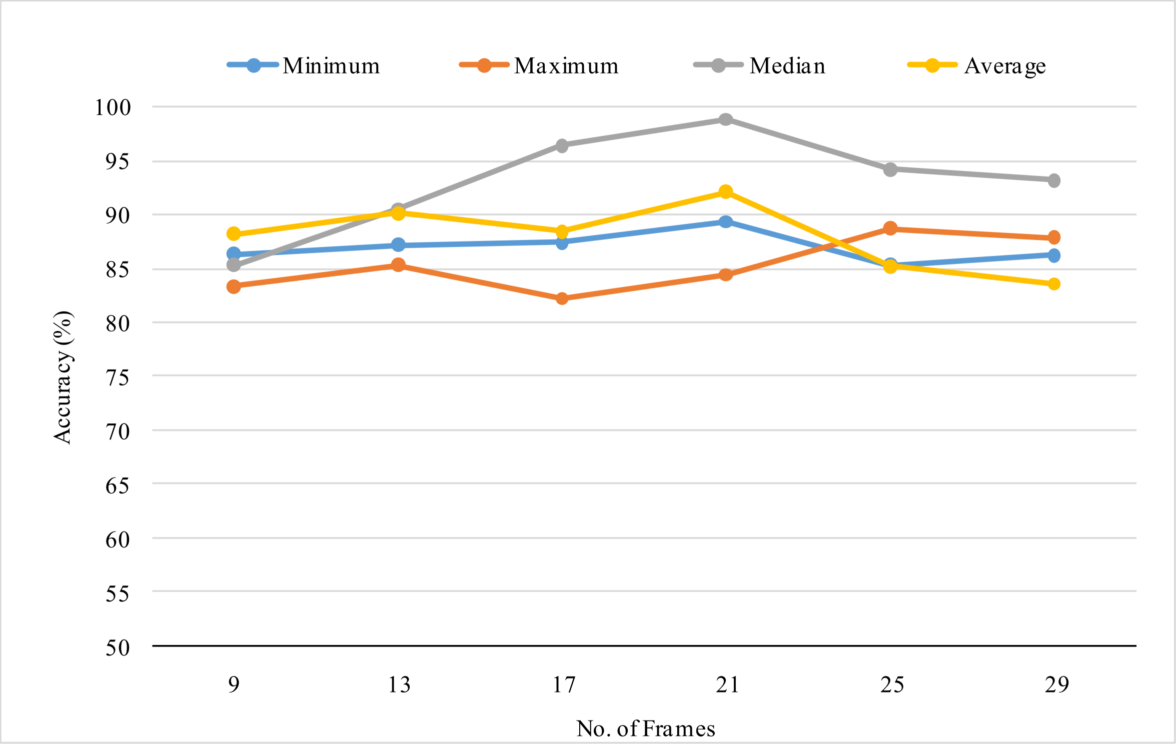


FIGURE 6. Accuracy (%) comparison of different aggregative functions with different video clips of size *W* frames.

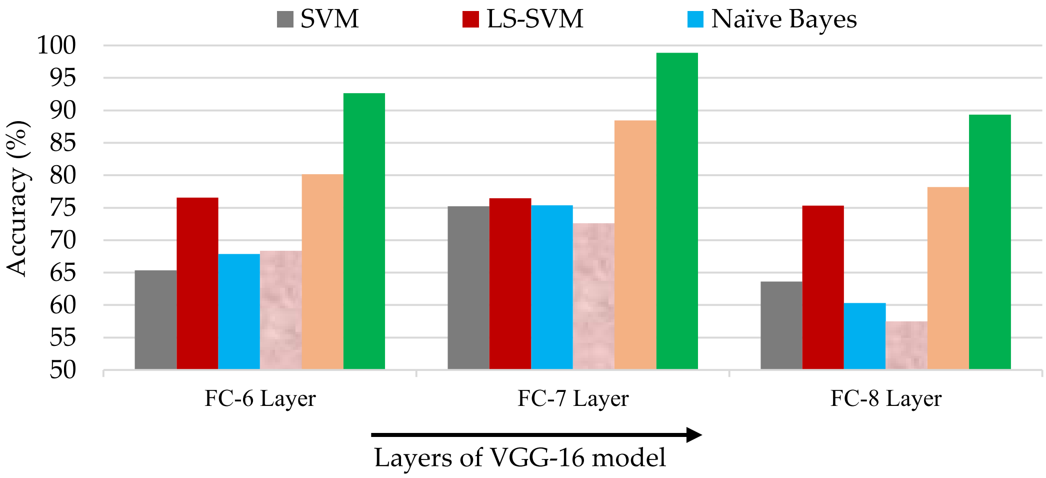


FIGURE 7. Effect of different layers of VGG-16 on accuracy with different classifiers.

it is evident from experiments that the best results are obtained with proposed parasitic layers using the RBF kernel as compared to other classifiers.

3) EFFECT OF DIFFERENT ACTIVATION AND KERNEL FUNCTIONS IN PARASITIC LAYERS

The accuracy depends on the number of hidden layer neurons and the way through which weights are calculated of these neurons. Two algorithms are applied as discussed earlier in the parasitic layers section. In the learning algorithm 1, the numbers of hidden layer neurons are selected manually, and weights are given randomly. Different activation functions are used to calculate the weights of output matrix **Z**. The experiments are started with the number of neurons 200 and are increased gradually with the step size 100 for all the activation functions to obtain the accuracy on dataset D7. For detail see Fig. 8 and Table 3.

It is evident from Fig. 8 that the accuracy is gradually increasing with an increasing number of neurons. The highest accuracy 90.33% is achieved using sigmoid function when the number of neurons are 700. As the number of neurons increased from 700 to 800, there is no change in the accuracy as shown in Fig. 8. Therefore, stopped further to tweak the

TABLE 3. Time efficiency and accuracy in percentage with different activation functions on dataset D7.

No. of Neurons	Sigmoid			Sine			Hand-limit		
	Accuracy	Tuning Time (s)	Testing Time (s)	Accuracy	Tuning Time (s)	Testing Time (s)	Accuracy	Tuning Time (s)	Testing Time (s)
800	90.33 ± 0.62	1.5313	0.0625	90.21 ± 0.93	1.6094	0.0625	89.78 ± 0.76	1.4688	0.0781
700	**90.33 ± 0.68**	**1.6875**	**0.0625**	**90.21 ± 0.72**	**1.3750**	**0.0625**	**89.78 ± 0.67**	**1.2656**	**0.0625**
600	88.69 ± 0.62	0.9219	0.0313	89.92 ± 0.82	1.6406	0.0625	88.53 ± 0.72	0.9375	0.0313
500	85.42 ± 0.55	0.7813	0.0313	87.19 ± 0.84	0.8594	0.0581	85.15 ± 0.68	0.7969	0.0313
400	81.47 ± 0.64	0.5469	0.0313	80.65 ± 0.92	0.7188	0.0313	82.70 ± 0.64	0.7813	0.0313
300	77.38 ± 0.75	0.4375	0.0313	74.25 ± 0.78	0.4375	0.0101	73.71 ± 0.62	0.7969	0.0313
200	69.62 ± 0.78	0.3281	0.0313	70.71 ± 0.87	0.4063	0.0010	70.16 ± 0.78	0.3438	0.0313

TABLE 4. Time efficiency and accuracy in percentage with different kernels on dataset D7.

Kernels	Parameters	Accuracy	Tuning Time (s)	Testing Time (s)	Kernels	Parameters	Accuracy	Tuning Time (s)	Testing Time (s)
RBF Kernel	C=4, σ=0.1	97.96 ± 0.72	0.3576	0.1223	Polynomial Kernel	C=3, b=0.2, d=1	90.19 ± 0.12	0.3734	0.1291
	C=4, σ=0.2	97.86 ± 0.82	0.4261	0.1030		C=3, b=0.2, d=2	91.55 ± 0.64	0.3921	0.1738
	C=4, σ=0.3	97.79 ± 0.59	0.3893	0.1666		C=3, b=0.2, d=3	93.73 ± 0.62	0.5145	0.1702
	C=3, σ=0.1	98.83 ± 0.52	1.5313	0.0758		C=3, b=0.2, d=4	50.54 ± 0.12	0.6291	0.1985
	C=3, σ=0.2	**98.89 ± 0.43**	**1.5313**	**0.0758**		C=3, b=0.3, d=1	90.19 ± 0.61	0.3621	0.1357
	C=3, σ=0.3	98.79 ± 0.58	0.1846	0.0753		C=3, b=0.3, d=2	92.64 ± 0.73	0.4030	0.1267
	C=2, σ=0.1	97.96 ± 0.81	0.2486	0.0970		**C=3, b=0.3, d=3**	**93.87 ± 0.43**	**0.5120**	**0.1931**
	C=2, σ=0.2	97.82 ± 0.53	0.4201	0.0787		C=3, b=0.3, d=4	50.95 ± 0.11	0.6649	0.1692
	C=2, σ=0.3	97.82 ± 0.77	0.2560	0.1002		C=3, b=0.4, d=1	90.19 ± 0.49	0.3731	0.1627
Linear Kernel	C=4	90.19 ± 0.25	0.4229	0.1835		C=3, b=0.4, d=2	92.78 ± 0.58	0.4193	0.1494
	C=3	**90.19 ± 0.34**	**0.3844**	**0.1700**		C=3, b=0.4, d=3	93.05 ± 0.54	0.3907	0.1552
	C=2	90.05 ± 0.32	0.4339	0.2033		C=3, b=0.4, d=4	51.50 ± 0.14	0.5608	0.2240
Polynomial Kernel	C=4, b=0.1, d=1	90.19 ± 0.15	0.3553	0.1398		C=3, b=0.5, d=1	90.19 ± 0.53	0.4256	0.1933
	C=4, b=0.1, d=2	91.83 ± 0.43	0.3807	0.1343		C=3, b=0.5, d=2	92.23 ± 0.51	0.4626	0.1737
	C=4, b=0.1, d=3	93.73 ± 0.57	0.5063	0.1524		C=3, b=0.5, d=3	93.19 ± 0.42	0.8246	0.1852
	C=4, b=0.1, d=4	50.95 ± 0.17	0.3752	0.0992		C=3, b=0.5, d=4	50.95 ± 0.12	1.4052	0.1722
	C=4, b=0.2, d=1	90.19 ± 0.18	0.3959	0.1622		C=2, b=0.1, d=1	90.05 ± 0.62	0.2184	0.0551
	C=4, b=0.2, d=2	91.69 ± 0.63	0.3018	0.1556		C=2, b=0.1, d=2	93.05 ± 0.75	0.1460	0.0517
	C=4, b=0.2, d=3	92.64 ± 0.62	0.3999	0.1609		C=2, b=0.1, d=3	93.46 ± 0.69	0.3893	0.1666
	C=4, b=0.2, d=4	50.54 ± 0.12	0.6677	0.1715		C=2, b=0.1, d=4	50.95 ± 0.17	0.3960	0.0562
	C=4, b=0.3, d=1	90.19 ± 0.12	0.4903	0.1667		C=2, b=0.2, d=1	90.05 ± 0.63	0.1537	0.0497
	C=4, b=0.3, d=2	92.51 ± 0.57	0.4047	0.1640		C=2, b=0.2, d=2	92.37 ± 0.68	0.1657	0.0492
	C=4, b=0.3, d=3	92.51 ± 0.57	0.5582	0.1588		C=2, b=0.2, d=3	94.14 ± 0.71	0.2099	0.0608
	C=4, b=0.3, d=4	50.95 ± 0.11	1.8708	0.1767		C=2, b=0.2, d=4	50.54 ± 0.15	0.2308	0.0650
	C=4, b=0.4, d=1	90.19 ± 0.16	0.2751	0.1638		C=2, b=0.3, d=1	90.05 ± 0.77	0.1424	0.0549
	C=4, b=0.4, d=2	92.51 ± 0.58	0.3356	0.1789		C=2, b=0.3, d=2	91.69 ± 0.62	0.1590	0.0556
	C=4, b=0.4, d=3	93.32 ± 0.44	0.5031	0.2029		C=2, b=0.3, d=3	93.60 ± 0.57	0.2367	0.0594
	C=4, b=0.4, d=4	51.50 ± 0.19	0.8039	0.1578		C=2, b=0.3, d=4	50.95 ± 0.13	0.2311	0.0560
	C=4, b=0.5, d=1	90.19 ± 0.15	0.5342	0.1871		C=2, b=0.4, d=1	90.05 ± 0.60	0.3970	0.2021
	C=4, b=0.5, d=2	92.37 ± 0.47	0.4841	0.2031		C=2, b=0.4, d=2	91.69 ± 0.68	0.3681	0.1881
	C=4, b=0.5, d=3	93.87 ± 0.52	0.6028	0.2314		C=2, b=0.4, d=3	93.46 ± 0.68	0.5019	0.1096
	C=4, b=0.5, d=4	50.95 ± 0.14	1.1659	0.1489		C=2, b=0.4, d=4	51.50 ± 0.20	0.6793	0.1568
	C=3, b=0.1, d=1	90.19 ± 0.18	0.4141	0.1202		C=2, b=0.5, d=1	90.05 ± 0.52	0.4997	0.2313
	C=3, b=0.1, d=2	92.23 ± 0.65	0.4034	0.1292		C=2, b=0.5, d=2	92.23 ± 0.45	0.3870	0.2012
	C=3, b=0.1, d=3	93.60 ± 0.72	0.4139	0.1421		C=2, b=0.5, d=3	93.27 ± 0.24	0.5958	0.2073
	C=3, b=0.1, d=4	50.95 ± 0.15	0.6947	0.1826		C=2, b=0.5, d=4	50.95 ± 0.12	1.1054	0.1719

values of neurons. Similarly, in the learning algorithm 2, different kernels (polynomial, linear and RBF) are used in the parasitic layers and achieved accuracies empirically on dataset D7, which are presented in Table 4. For the polynomial kernel, three parameters (C, b, and d) needs to be tuned to obtain accuracy. The best accuracy 93.87% is achieved

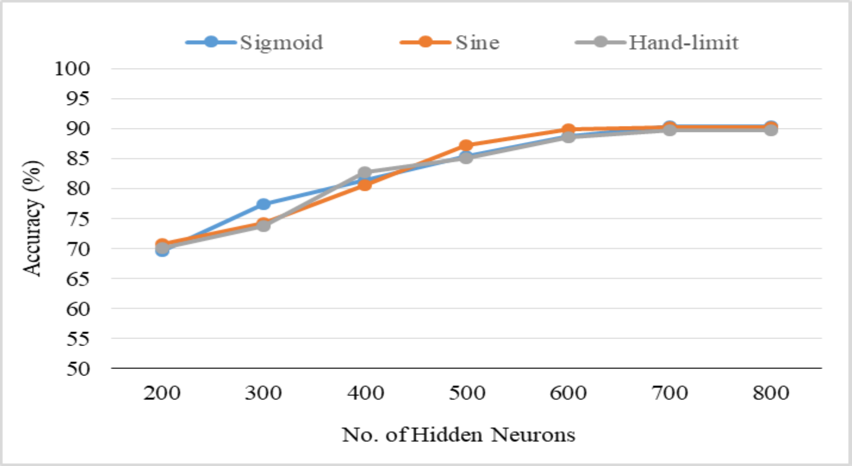
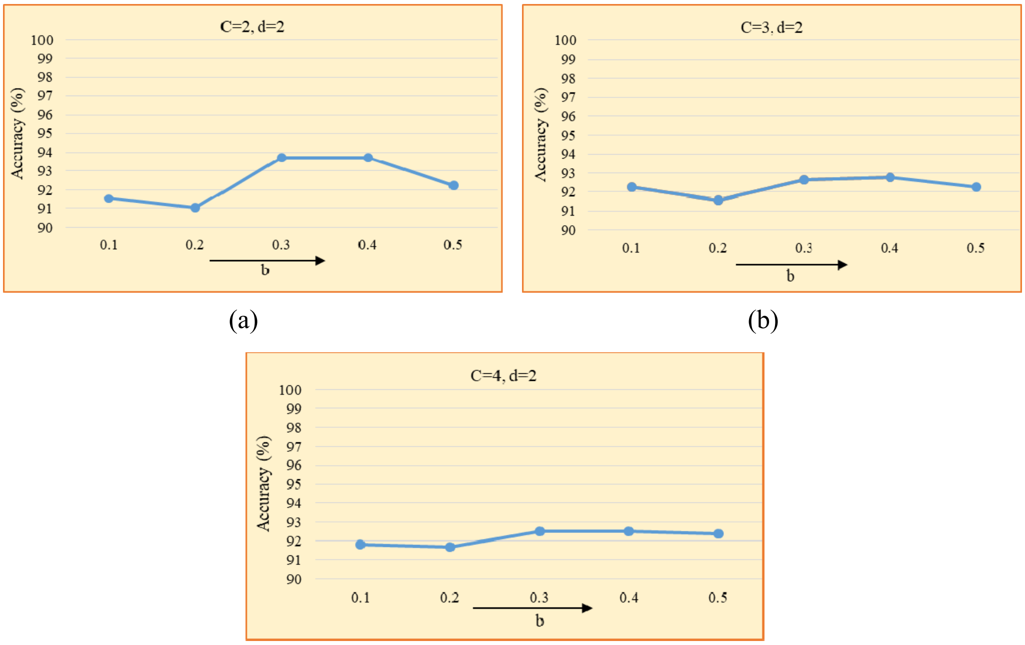


FIGURE 8. Effect of different number of hidden neurons.



(a)

(b)

(c)

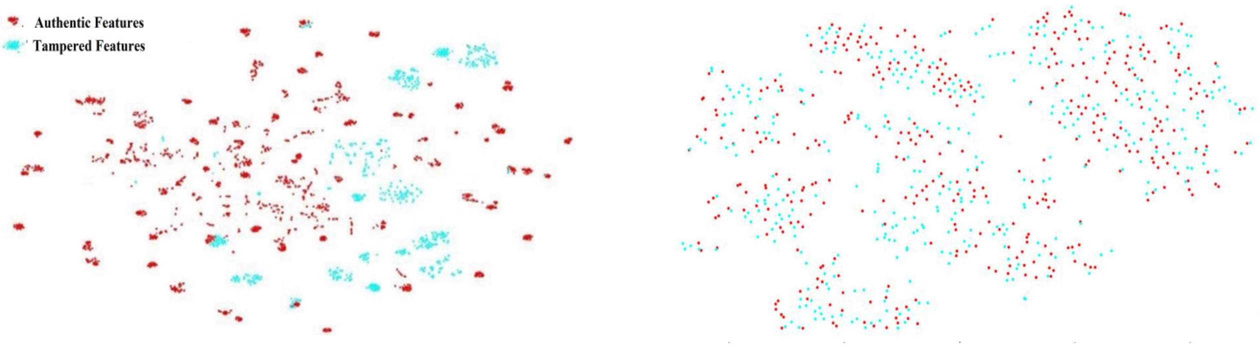
FIGURE 9. Effect of parameter value b in polynomial kernel on accuracy. (a) When $C = 2$, $d = 2$. (b) When $C = 3$, $d = 2$. (c) When $C = 4$, $d = 2$.

on the polynomial kernel with the parameter values $C = 3$, $b = 0.3$, and $d = 3$. The values of parameters (C, b, and d) increased gradually until accuracy stopped increasing further. The parameter b is started tweaking from 0.1 with the increment of step size 0.1. A graph is shown in Fig. 9 for different values of b (0.1 to 0.5) at ($C = 2, 3, 4$) and $d = 2$. The accuracy started decreasing after $b = 0.4$, therefore, stopped tweaking further. A similar, process is also completed with the value of d *and* C. The highest accuracy 90.16% is achieved for the linear kernel at $C = 3$. For further improvement in accuracy, the RBF kernel is investigated and to get results the values of parameters (C and σ) are tuned empirically. The best accuracy 98.89% is obtained with $C = 3$, $\sigma = 0.2$ on this kernel because this kernel has good generalization capabilities and tolerance to noise.

The time efficiency is also calculated for both algorithms. The results with time calculation are shown in Table 3 and 4. The time efficiency is improved which is also endorsed during comparison with methods [8] and [23] (see Table 6). The reason to improve time efficiency is elimination of gradient descent algorithm to calculate weights of the neurons in FC8 layer. Moreover, the method learns the parameters of FC8 and FC9 by solving the closed form optimization problem.

B. EFFECTIVENESS ON DIFFERENT DATASET

The proposed method is verified on different tampered video datasets. The achieved accuracy is 96.34%, 97.45%, 95.37%, 96.75%, 93.68% and 98.89% on D2, D3, D4, D5, D6, and



(a)

(b)

FIGURE 10. (a) Visualization of features distribution extracted through FC7 of proposed method. (b) Visualization of features distribution of [23].

TABLE 5. Performance of the proposed method on various datasets and their combination.

Sr. No.	Datasets	TPR (Recall) (%)	TNR (Specificity) (%)	AC (Accuracy)(%)
1	D2	94.28	93.45	91.34 ± 0.98
2	D3	98.37	97.23	97.45 ± 0.59
3	D4	96.81	94.56	95.37 ± 0.79
4	D5	98.45	96.78	96.75 ± 0.76
5	D6	96.51	95.34	93.68 ± 0.56
6	**D7**	**99.12**	**97.25**	**98.89** ± **0.43**

D7 datasets respectively as shown in Table 5. The best accuracy 98.89% is accomplished on D7, which is a combination of all datasets. The lowest and highest accuracies are 93.68% and 98.89% on dataset D6 and D7 respectively. The accuracies on dataset D6 is the lowest because in this dataset the authentic and forged videos were not the same. Furthermore, the proposed method is independent of frame rate and video resolution. Each video is divided into non-overlapping VCs of fixed size $W = 21$ frames which makes method independent of frame rate, and then checks the authenticity of each VC instead of verification of video. Similarly, the method takes VC of any resolution and MR layer gives the rescaled output of 224 × 224 × 3, due to this reason the method works on all resolution.

C. COMPARISON WITH THE STATE-OF-THE-ART

For validation the results are compared with state-of-the-art methods (see Table 6 and Table 7). To the best of our knowledge, only a few methods are available in the literature for detection of object-based tampered video. The proposed method is compared with two state-of-the-art techniques dealing with object-based tampering in videos developed by Richao *et al.* [8] and Chen *et al.* [23]. The results achieved with proposed method and state-of-the-art techniques on dataset D7 are shown in Table 6.

The comparison reveals that the proposed method has better accuracy on D7 dataset, which involves different types of geometric transformations and post processing operations. The method outperforms than the both methods and works well against different types of geometric transformations and post-processing operations. The success of any classification system depends on how accurately? it models the structural changes occurring in video frames due to tampering. During tampering lines, edges, and corners are introduced, which are considered artifacts of tampering. Feature extraction using modified CNN represented these artifacts more precisely and the classification results are improved. For a fair comparison, the method requires to be compared on same dataset. Since, the videos used in [8] and [23] are not publicly available, and the proposed method cannot be tested on these videos. The methods in [8] and [23] were implemented with same setting of parameters and were tested on dataset D7. The comparison is shown in Table 6. The performance of the proposed method is significant as compared to methods in [8] and [23] because hierarchical features extracted using CNN layers are more discriminative than hand-crafted features.

The performance of the proposed method is also invariant to different geometric transformations such as scaling, rotation, shearing and mirroring as compared to methods in [1] and [26]. The accuracy of the proposed method is better than the state-of-the-art methods which are trained and tested on D2 dataset, as shown in Table 7.

V. DISCRIMINATIVE ANALYSIS OF FEATURES

The extracted features from FC7 layer of the proposed method are also evaluated to study the discriminating nature of the features. The method of t-SNE [53] is introduced as the evaluation criteria. This method is commonly used for evaluating the discriminating properties of features [54]. The visualization of features distribution extracted from FC7 layer of the proposed method and method in [23] is shown in Fig. 10. Two coloured points represent instances of two types. Red points represent the authentic instances and the blue points represent tampered instances. It is obvious to see from Fig. 10a that the proposed method gives discriminating high level features which contribute to provide good accuracy and better generalization as compared to [23].

TABLE 6. Comparison of the proposed method and the state-of-the-art methods on dataset D7.

Methods	TPR (Recall) (%)	TNR (Specificity) (%)	AC (Accuracy) (%)	Testing Time (s)
Proposed Method	**99.12**	**97.25**	**98.89 ± 0.43**	**0.0758**
Method in [23]	68.31	70.34	72.67 ± 1.08	0.2898
Method in [8]	59.42	61.23	63.6 ± 0.98	0.1482

TABLE 7. Comparison of the proposed method in terms of accuracy with state-of-the-art methods on dataset D2.

Methods	AC (Accuracy) (%)	Invariant to scaling, rotation, shearing and mirror
Proposed Method	**96.34**	**All**
Method in [1]	93.40	All
Method in [8]	59.20	No
Method in [24]	89.70	No
Method in [23]	68.32	No
Method in [26]	92.60	Mirror

VI. CONCLUSION AND FUTURE WORK

Detection of altered videos is a challenging task. The current cutting-edge techniques face limitations ranging from evaluation of the method on a small number of videos, to use of single video formats and a fixed resolution. In this paper, a method based on deep model is proposed for classification of authentic and tampered video clips (VCs) which consists of three types of layers i.e., motion residual, CNN, and parasitic layers. Residual layer aggregates the information from consecutive frames of VC, CNN layers extract discriminating features from the aggregated frame of a VC and parasitic layers employed these features to classify a VC as authentic or tampered. The parasitic layers are computationally efficient and learn the structure of authentic and tampered VCs efficiently. The proposed method gives better accuracy 98.89% and also computationally efficient. The method can detect even the most plausibly created forgery. Moreover, the proposed method may motivate the research community to think another way to enhance the performance of convolution neural networks in the domain of tampering detection. Our future work will be focused on the localization of tampered objects in the video clips.

REFERENCES

[1] M. Saddique, K. Asghar, U. I. Bajwa, M. Hussain, and Z. Habib, "Spatial video forgery detection and localization using texture analysis of consecutive frames," *Adv. Electr. Comput. Eng.*, vol. 19, no. 3, pp. 97–108, 2019.

[2] M. Sadddique, K. Asghar, T. Mehmood, M. Hussain, and Z. Habib, "Robust video content authentication using video binary pattern and extreme learning machine," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 8, pp. 264–269, 2019.

[3] J. Tao, L. Jia, and Y. You, "Review of passive-blind detection in digital video forgery based on sensing and imaging techniques," in *Proc. Int. Conf. Optoelectron. Microelectron. Technol. Appl.*, Shanghai, China, Jan. 2017, Art. no. 102441.

[4] S. Jia, Z. Xu, H. Wang, C. Feng, and T. Wang, "Coarse-to-fine copy-move forgery detection for video forensics," *IEEE Access*, vol. 6, pp. 25323–25335, 2018.

[5] K. Asghar, Z. Habib, and M. Hussain, "Copy-move and splicing image forgery detection and localization techniques: A review," *Austral. J. Forensic Sci.*, vol. 49, no. 3, pp. 281–307, May 2017.

[6] K. Asghar, X. Sun, P. L. Rosin, M. Saddique, M. Hussain, and Z. Habib, "Edge–texture feature-based image forgery detection with cross-dataset evaluation," *Mach. Vis. Appl.*, vol. 30, nos. 7–8, pp. 1243–1262, Oct. 2019.

[7] C.-M. Pun, X.-C. Yuan, and X.-L. Bi, "Image forgery detection using adaptive oversegmentation and feature point matching," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 8, pp. 1705–1716, Aug. 2015.

[8] C. Richao, Y. Gaobo, and Z. Ningbo, "Detection of object-based manipulation by the statistical features of object contour," *Forensic Sci. Int.*, vol. 236, pp. 164–169, Mar. 2014.

[9] M. Kobayashi, T. Okabe, and Y. Sato, "Detecting video forgeries based on noise characteristics," in *Advances Image Video Technology*, vol. 5414. Tokyo, Japan: Springer, 2009, pp. 306–317.

[10] D.-K. Hyun, S.-J. Ryu, H.-Y. Lee, and H.-K. Lee, "Detection of upscale-crop and partial manipulation in surveillance video based on sensor pattern noise," *Sensors*, vol. 13, no. 9, pp. 12605–12631, 2013.

[11] R. C. Pandey, S. K. Singh, and K. K. Shukla, "Passive copy-move forgery detection in videos," in *Proc. Int. Conf. Comput. Commun. Technol. (ICCCT)*, Allahabad, India, Sep. 2014, pp. 301–306.

[12] J. Goodwin and G. Chetty, "Blind video tamper detection based on fusion of source features," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl.*, Noosa, QLD, Australia, Dec. 2011, pp. 608–613.

[13] C.-C. Hsu, T.-Y. Hung, C.-W. Lin, and C.-T. Hsu, "Video forgery detection using correlation of noise residue," in *Proc. IEEE 10th Workshop Multimedia Signal Process.*, Cairns, Qld, Australia, Oct. 2008, pp. 170–174.

[14] L. Su, C. Li, Y. Lai, and J. Yang, "A fast forgery detection algorithm based on exponential-Fourier moments for video region duplication," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 825–840, Apr. 2018.

[15] C. Guo, Y. Zhu, and G. Luo, "A detection method for facial expression reenacted forgery in videos," in *Proc. 10th Int. Conf. Digit. Image Process. (ICDIP)*, Shanghai, China, Aug. 2018, Art. no. 108061.

[16] M. A. Bagiwa, A. W. A. Wahab, M. Y. I. Idris, S. Khan, and K.-K.-R. Choo, "Chroma key background detection for digital video using statistical correlation of blurring artifact," *Digit. Invest.*, vol. 19, pp. 29–43, Dec. 2016.

[17] R. D. Singh and N. Aggarwal, "Detection of upscale-crop and splicing for digital video authentication," *Digit. Invest.*, vol. 21, pp. 31–52, Jun. 2017.

[18] G. Singh and K. Singh, "Video frame and region duplication forgery detection based on correlation coefficient and coefficient of variation," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 11527–11562, May 2019.

[19] A. Bidokhti and S. Ghaemmaghami, "Detection of regional copy/move forgery in MPEG videos using optical flow," in *Proc. Int. Symp. Artif. Intell. Signal Process. (AISP)*, Mashhad, Iran, Mar. 2015, pp. 13–17.

[20] L. Li, X. Wang, W. Zhang, G. Yang, and G. Hu, "Detecting removed object from video with stationary background," in *Proc. Int. Workshop Digit. Forensics Watermarking*, Taipei, Taiwan, 2013, pp. 242–252.

[21] O. I. Al-Sanjary, A. A. Ahmed, A. A. B. Jaharadak, M. A. M. Ali, and H. M. Zangana, "Detection clone an object movement using an optical flow approach," in *Proc. IEEE Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, Penang, Malaysia, Apr. 2018, pp. 388–394.

[22] P. Bestagini, S. Milani, M. Tagliasacchi, and S. Tubaro, "Local tampering detection in video sequences," in *Proc. IEEE 15th Int. Workshop Multimedia Signal Process. (MMSP)*, Pula, CA, Italy, Sep. 2013, pp. 488–493.

[23] S. Chen, S. Tan, B. Li, and J. Huang, "Automatic detection of object-based forgery in advanced video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 11, pp. 2138–2151, Nov. 2016.

[24] A. V. Subramanyam and S. Emmanuel, "Video forgery detection using HOG features and compression properties," in *Proc. IEEE 14th Int. Workshop Multimedia Signal Process. (MMSP)*, Banff, AB, Canada, Sep. 2012, pp. 89–94.

[25] J. Zhang, K. Huang, Y. Yu, and T. Tan, "Boosted local structured HOG-LBP for object localization," in *Proc. CVPR*, El Paso, CA, USA, Jun. 2011, pp. 1393–1400.

[26] L. Su and C. Li, "A novel passive forgery detection algorithm for video region duplication," *Multidimensional Syst. Signal Process.*, vol. 29, no. 3, pp. 1173–1190, Jul. 2018.

[27] L. Su, T. Huang, and J. Yang, "A video forgery detection algorithm based on compressive sensing," *Multimedia Tools Appl.*, vol. 74, no. 17, pp. 6641–6656, Sep. 2015.

[28] V. Conotter, J. F. O'Brien, and H. Farid, "Exposing digital forgeries in ballistic motion," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 283–296, Feb. 2012.

[29] Y. Yao, Y. Shi, S. Weng, and B. Guan, "Deep learning for detection of object-based forgery in advanced video," *Symmetry*, vol. 10, no. 1, p. 3, 2017.

[30] M. Zampoglou, F. Markatopoulou, G. Mercier, D. Touska, E. Apostolidis, S. Papadopoulos, R. Cozien, I. Patras, V. Mezaris, and I. Kompatsiaris, "Detecting tampered videos with multimedia forensics and deep learning," in *Proc. Int. Conf. Multimedia Model.*, Thessaloniki, Greece, 2019, pp. 374–386.

[31] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief net model for visual area V2," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2008, pp. 873–880.

[32] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin, "Exploring strategies for training deep neural networks," *J. Mach. Learn. Res.*, vol. 10, no. 1, pp. 1–40, Jan. 2009.

[33] J. Chen, X. Kang, Y. Liu, and Z. J. Wang, "Median filtering forensics based on convolutional neural networks," *IEEE Signal Process. Lett.*, vol. 22, no. 11, pp. 1849–1853, Nov. 2015.

[34] Y. Zhan, Y. Chen, Q. Zhang, and X. Kang, "Image forensics based on transfer learning and convolutional neural network," in *Proc. 5th ACM Workshop Inf. Hiding Multimedia Secur. IHMMSec*, Philadelphia, PA, USA, 2017, pp. 165–170.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2012, pp. 1097–1105.

[37] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1120–1124, Sep. 2014.

[38] X. Cao, Z. Wang, P. Yan, and X. Li, "Transfer learning for pedestrian detection," *Neurocomputing*, vol. 100, pp. 51–57, Jan. 2013.

[39] D. Wu, F. Zhu, and L. Shao, "One shot learning gesture recognition from RGBD images," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Providence, RI, USA, Jun. 2012, pp. 7–12.

[40] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, Zurich, Switzerland, 2014, pp. 818–833.

[41] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Boston, MA, USA, Jun. 2015, pp. 36–45.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[43] C. Combes, *Parasitism: The Ecology and Evolution of Intimate Interactions*. Chicago, IT, USA: University of Chicago Press, 2001.

[44] L. Zheng, T. Sun, and Y.-Q. Shi, "Inter-frame video forgery detection based on block-wise brightness variance descriptor," in *Proc. Int. Workshop Digit. Watermarking*, Tokyo, Japan, 2014, pp. 18–30.

[45] A. B. Sargano, X. Wang, P. Angelov, and Z. Habib, "Human action recognition using transfer learning with deep representations," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, Alaska, May 2017, pp. 463–469.

[46] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[47] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Paris, France: Springer, 2010, pp. 177–186.

[48] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 42, no. 2, pp. 513–529, Apr. 2012.

[49] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006.

[50] G. Qadir, S. Yahaya, and A. T. S. Ho, "Surrey university library for forensic analysis (SULFA) of video content," in *Proc. IET Conf. Image Process. (IPR)*, London, U.K., 2012, pp. 1–6.

[51] E. Ardizzone and G. Mazzola, "A tool to support the creation of datasets of tampered videos," in *Proc. Int. Conf. Image Anal. Process.*, Genova, Italy, 2015, pp. 665–675.

[52] O. Ismael Al-Sanjary, A. A. Ahmed, and G. Sulong, "Development of a video tampering dataset for forensic investigation," *Forensic Sci. Int.*, vol. 266, pp. 565–572, Sep. 2016.

[53] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[54] Z. Jiao, X. Gao, Y. Wang, and J. Li, "A parasitic metric learning net for breast mass classification based on mammography," *Pattern Recognit.*, vol. 75, pp. 292–301, Mar. 2018.

MUBBASHAR SADDIQUE received the B.Sc. degree in telecommunication engineering from the Institute of Engineering and Technology, Lahore Campus, Pakistan, and the M.S. degree in computer science from COMSATS University Islamabad (Abbottabad Campus), Pakistan, in 2010. He has also worked as a Research Associate at the Department of Cyber Defense Graduate, School of Information Security, Korea University, South Korea. He is currently a Ph.D. Scholar at COMSATS University Islamabad, Pakistan. He is also working as a Lecturer with the Department of Computer Sciences, University of Okara, Pakistan. He is currently working in video and image forensic domain. Furthermore, his research interests are in the area of image/video processing, computer vision, machine learning, data mining, and networks. He received his merit scholarship from COMSATS University Islamabad (Abbottabad Campus).

KHURSHID ASGHAR received the Ph.D. degree in the field of image forensics from COMSATS University Islamabad, Pakistan. He is also working as an Associate Professor with the Department of Computer Science, University of Okara. He has also worked as a Research Associate at the Cardiff School of Computer Science and Informatics, Cardiff University, U.K. His current research interests include image processing, image and video forensics, machine learning, deep learning, network security, bio-metrics, medical imaging brain signals, geometric modeling, and computer graphics.

USAMA IJAZ BAJWA received the M.S. and Ph.D. degrees in computer engineering from the Center for Advanced Studies in Engineering Islamabad, Pakistan, in 2006 and 2013, respectively. He is an aggressively enterprising person, who is currently an Assistant Professor, Research Supervisor, failed entrepreneur, Program Chair of International Conference on FIT, the Associate Head of the Computer Science Department, COMSATS University Islamabad (CUI) Lahore Campus, and a polyglot. Previously, he was an Assistant Professor and Graduate Program Coordinator at CUI Abbottabad, and the Head of the Information Security and Image Processing Research Group at CUI Abbottabad, the Team Lead in ICT R&D funded Project, the Co-Founder of Technology Nucleus Pvt., Ltd., and a Visiting Researcher at the Medical Imaging Lab, University of South Wales. He has authored/coauthored 40 international conference and journal research papers. His research interests include biometrics, medical image analysis, image processing, computer vision, and pattern classification.

MUHAMMAD HUSSAIN received the M.Sc. and M.Phil. degrees from the University of the Punjab, Lahore, Pakistan, in 1990 and 1993, respectively, and the Ph.D. degree in computer science from Kyushu University, Fukuoka, Japan, in 2003. He is currently a Professor with the Department of Computer Science, King Saud University, Saudi Arabia. His current research interests include image and signal processing, pattern recognition, and computer graphics. His research has been funded by the Japan Science and Technology Agency (JST), and the National Science Technology and Innovation Plan (NSTIP) of Saudi Arabia. He was an Editor of the *Journal of Computer and Information Sciences* (Elsevier), King Saud University, and has served on the program committees of various international conferences.

HATIM A. ABOALSAMH received the Ph.D. degree in computer engineering and science from the University of Miami, Miami, FL, USA, in 1987. He is currently a Professor and Chairman of the Department of Computer Science, King Saud University, Saudi Arabia. His research interests include pattern recognition, biometrics, probability modeling, and machine learning. He is currently the Editor-in-Chief of the *Saudi Computer Society Journal*, and a Fellow Member of the British Computer Society (BCS). He is a Senior Member of the Association of Computing Machinery, USA (ACM), a Senior Member of the International Association of Computer Science and Information Technology (IACSIT), Singapore, and a Professional Member of IEEE. He held leading posts of Vice Rector for Development and Quality-King Saud University (KSA), Riyadh, Saudi Arabia, from 2006 to 2009, and the Dean of the College of Computer and Information Sciences, the Editor-in-Chief of the *KSU-Journal of Computer Sciences*, and the Vice President of the Saudi Computer Society.

ZULFIQAR HABIB was born in Pakistan, in September 1966. He received the Ph.D. degree in computer science from Kagoshima University, Japan, in 2004, followed by the Award of Postdoctoral Fellowship of two years by the Japan Society for the Promotion of Science. He has been holding the position of Full Professor, since 2012, and the founding In-Charge of the Center for Research in VIG-Robotics, Department of Computer Science, COMSATS University Islamabad (CUI), Lahore Campus. He has also served as the Chairman for eight campuses of CUI. His teaching and research interests broadly span the areas of digital image processing, computer graphics, computer vision, machine learning, and robotics. He has published a large number of research papers in impact factor journals of good JCR. He has supervised Ph.D. students with the successful completion of their degrees. Most of the graduate students are completing their degrees in the normal duration and are being sponsored through local grants and International funding projects, like Horizon 2020. He has achieved various awards in education and research, including two Research Productivity Awards by the Ministry of Science and Technology, Pakistan, and three graduate merit fellowships by the Japanese and German Governments. He is engaged in many academic activities, an organizer of various international conferences, and a member or advisor of editorial/academic boards. Since 2009, he has been invited to give keynote lectures or tutorial in numerous national and International conferences and as a Guest Researcher in the universities of Turkey and U.K.