Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.010869
Article

Tech Science Press

A Novel Forgery Detection in Image Frames of the Videos Using Enhanced Convolutional Neural Network in Face Images

S. Velliangiri[1,*] and J. Premalatha[2]

[1]CMR Institute of Technology, Hyderabad, 501401, India
[2]Kongu Engineering College, Erode, 638052, India
*Corresponding Author: S. Velliangiri. Email: velliangiris@gmail.com
Received: 02 April 2020; Accepted: 14 June 2020

Abstract: Different devices in the recent era generated a vast amount of digital video. Generally, it has been seen in recent years that people are forging the video to use it as proof of evidence in the court of justice. Many kinds of researches on forensic detection have been presented, and it provides less accuracy. This paper proposed a novel forgery detection technique in image frames of the videos using enhanced Convolutional Neural Network (CNN). In the initial stage, the input video is taken as of the dataset and then converts the videos into image frames. Next, perform pre-sampling using the Adaptive Rood Pattern Search (ARPS) algorithm intended for reducing the useless frames. In the next stage, perform pre-processing for enhancing the image frames. Then, face detection is done as of the image utilizing the Viola–Jones algorithm. Finally, the improved Crow Search Algorithm (ICSA) has been used to select the extorted features and inputted to the Enhanced Convolutional Neural Network (ECNN) classifier for detecting the forged image frames. The experimental outcome of the proposed system has achieved 97.21% accuracy compared to other existing methods.

Keywords: Adaptive Rood Pattern Search (ARPS); Improved Crow Search Algorithm (ICSA); Enhanced Convolutional Neural Network (ECNN); Viola Jones algorithm Speeded Up Robust Feature (SURF)

1 Introduction

The rapid development in video editing application has ended video forgery an easy task. Hence, the trustworthiness of hypermedia matters, particularly videos, as a proof is a cumbersome task. The growth of images processing software and the progression in cameras (digital) has brought about a significant amount of doctored images with no noticeable traces, creating a high demand for automatic forgery detection algorithms to ascertain the honesty of a candidate image [1]. Nowadays, it produces considerable difficulty in authenticating images. Image forgery implies the manipulation of the image (digital) to cover some essential or helpful information as of it [2]. Many research works focused on image forgery detection (IFD). The disclosure of the forgery algorithm will depend on the image source [3].

Forgery detection could well be developed as active or passive [4]. Active approaches were traditionally utilized by engaging data hiding (watermarking) or digital signatures [5]. Usually, the image watermarking is either embedded at the interval [6] of the time of the image acquisition or advanced after further processing of the real image. Contrary to active approaches, passive approaches do not count on pre-registration or pre-embedded information, and no meticulous research has been done on them. Passive methods for image forensic work on the dearth of any signature or water-mark [7].

Prior research has exhibited that copy-paste forgery can well be detected via finding localized inconsistencies in intrinsic image features such as traces of re-sampling, JPEG compression [8], median filtering [9], contrast enhancement, along with sensor noise. The forgery detection's output could well be of '2' types: (a) classifying the image as genuine or else forged (no localization), (b) localizing the forged region, on the off-chance of the image is not genuine [10]. Mainly '2' classes of CMFD are presented. One is centered on block-wise division and the next one on key-point extraction [11].

Every forgery detecting methods pursue a specific pattern, i.e., feature extraction, matching and post-processing. Since its last century, object features detectors are quite well-liked devices throughout the area of computer vision. They are incorporated in different applications such as object representation, object detection and matching, image recognition and recovery, 3D scene creation, activity recognition, text classification and biometric systems [12]. To encourage this, video forgery ought to develop either adequate information to streamline the production of precise altering veils or discharge the covers themselves. Moreover, video information ought to be dispersed to limit further preparation. Video preparing, for example, compression and correcting, can viably disguise altering. While the location of such an enemy of crime scene investigation is a significant research course, handling can be applied to video freely after dataset distribution, yet just if the first dataset is distributed to evade unnecessary handling. Much deep learning method has been proposed to solve this issue. As the assortment of video control procedures extends and propels, the altered and engineered video will get unclear from the original video to human eyes. Based on the review of different approaches, new procedures are required which either arrange altered video agreeing to its altering type or perform altering identification independent of the kind of altering. Keep up trust in the credibility of video content in the future, and it is vital to create procedures that can distinguish and limit video handling and control. For the most part, the techniques created to address one sort of tampering are not fit for tending to different sorts of imitations. The presentation of this technique relies upon the codecs utilized for compression and the video content. For instance, strategies fit for identifying tampering in quick movement recording will fail in slow movement recordings, and static foundations will bomb in dynamic foundation recordings and so forth.

The paper structure is organized as follows. Section 2 surveys the associated works regarding the method proposed. In sections 3, a concise discussion about the proposed methodology is given, section 4, explore the Investigational outcome, and also Section 5 concludes the paper with future directions.

2 Related Work

Hu et al. [13] suggested an IFD system for efficiently identifying a tampered background or foreground image utilizing image watermarking along with alpha mattes. This approach had '2' parts: (i) watermark embedding and (ii) identification of tempered images. The component-hue-difference-centered spectral matting was first utilized to attain the alpha matte. Subsequently, DWT-DCT-SVD-centered image watermarking was utilized to include the watermarks. Lastly, the difference betwixt the attained singular values was utilized for detecting the tempered background and foreground images.

Bhartiya et al. [14] presented a technique to detect sham on the JPEG image. An algorithm was designed to categorize the image blocks as non-forged or forged centered on an exacting feature that existed in multiple-compressed [JPEG] images. This approach modeled the characters present in the histograms of double compressed JPEG images for detecting forgery using feature-based clustering. The method performed was superior to the prior works that used the probability centered system for detecting a forgery on JPEG images. This approach showed the accuracy centered on quantitatively and qualitatively analysis only.

Mahmood et al. [15] recommended a robust method aimed at Copy-Moves Forgery detection (CMFD) along with localization on digital images. This approach consists of three steps: pre-processing, FE as of the image blocks, feature matching, along with filtering. In the pre-processing, the inputted image (RGB) was transformed into YC_bC_r the FE phase, and the image was split into blocks. In feature matching, the process was performed for image block pairs, and the post-processing was performed via morphological operation.

Elsharkawy et al. [16] presented an efficient blind IFD algorithm. Primarily, the image was attained with digital cameras together with scanned images. Subsequently, the logarithmic operation was applied to the input image for obtaining the illumination and also reflectance elements. Afterward, the low-pass filter image's histogram was estimated. Then, differentiation was implemented to the attained histogram for detecting the changes in illumination. Lastly, Support Vectors Machine (SVM) was utilized for training and testing to attain the feature vector. The Comparison outcomes proves the performance of this system.

Oommen et al. [17] offered a blind or passive approach to CMFD. This approach has seven steps, preprocessing, FE with LFD, image segmentation, organize segment in B+ Tree, Estimate SVD, Image blocks matching using SVD, and Filtering along with highlighting. The FE was done by using Local Fractals Dimension (LFD), and the LFD was estimated using Differential Box-Counting. The block matching was applied centered on Singular Values Decomposition (SVD). Youseph et al. [18] introduces techniques of detecting forged images using illuminant color assessment. It helps to detect the canny edges and extracts the shape features by HOG edge. Moreover, this technique has advantages of a minimum quantity of human interaction and improved the accuracy performance.

Pun et al. [19] propounded a CMF scheme utilizing adaptive over-segmentation and features-point matching. Initially, this was implemented for segmenting the [host] image into non-overlapping. Subsequently, the Scale Invariants Feature Transforms (SIFT) was applied in every block to extort the SIFT feature points. Lastly, the Forgery Region Extractions method was used for detecting the fake region as of the host image as per the extorted Labeled Features Points (LFP).

Zhong et al. [20] presented an enhanced block-centered efficient technique for CMFD. Initially, the suspicious image was inputted to pre-processing. Next, the step was the block

segmentation. The image was split into the overlapped circular block. Then, the features vector of every block was extorted by utilizing Discrete Radial Harmonics (DRHMs). The SVD and 2 Nearest Neighbors (2NN) was implemented for block matching. Lastly, the forgery regions were signified by white pixels.

Barani et al. [21,22] designed a digital image tamper detection algorithm based on integer wavelet transform. Proposed techniques recognize altered locales in different pictures, and furthermore, the technique has high inserting visual quality. The proposed technique assessed on various picture datasets and the outcomes as far as location execution are adequate, yet in individual assaults, the confirmation strategy has high recognition errors. These errors are self-explanatory in individual assaults, for example, JPEG pressure at high-pressure rates, what is more, salt and pepper at low commotion rates.

Hong et al. [23] recommended a detection of frame deletion in HEVC-Coded video. The proposed technique is triple. To begin with, it examines and recognizes video phony in the packed space, making it straightforward also, quick. Second, it arranges video as certifiable or produced without requiring recognition of the creases created upon outline erasure, making it hearty against a crease concealing handling. The trial results show that the proposed method altogether beats the current strategies and stays hearty in different erasure circumstances.

Uliyan et al. [24] reported the Anti-spoofing method for fingerprint recognition using a patch-based deep learning machine, which discusses with complex surface examples in a friendly manner because of its probabilistic multilayer engineering. KNN classifier is applied with the component vectors of the ROIs removed by the DBM to look at parody fabrications. The examination results show that the Deep learning model is hearty against various types of parody imitations. The exhibition assessment of the DRBM + DBM strategy accomplished best in class brings about three open unique mark acknowledgment benchmarks.

Bi et al. [25,26] proposed a quick duplicate move fraud detection using local bidirectional coherency error refinement. The proposed technique can hold excellent execution under various phony situations. Duplicate move falsification identification approach, setting up highlight correspondence under various assaults, is cultivated very well by the upgraded coherency delicate-looking.

3 Proposed Methodology

3.1 Forgery Detection in Image Frames of the Videos

Rapid development in digitalization causes spread of image along with video processing tools, such as Photoshop, Adobe Premiere, and also Final Cut Pro makes it easy to manipulate with digital visual media without any changes in existence. Nowadays, most of the legal and social issues were probably might occur due to malicious tampering. Major portions are tampered images and recordings used for false evidence in the trial and spread the rumors or false news among the people about political leaders or mislead the information. Meanwhile, the huge volume of digital knowledge makes it more difficult to identify interference through empirical observation alone. For detection, this work proposed forgery detection on image frames of the videos using Enhanced CNN. This work takes the Youtube8M as the dataset. Initially, the inputted video is transmuted into frames, and then the frames are pre-sampled for reducing the useless frames using ADPS. Preprocessing is performed for enhancing the image frames. Next, the features are extracted, and then the necessary features are selected as of the extorted features using ICSA.

Finally, the image frame is classified centered on the chosen features using ECNN. The structural design of the proposed work is shown in Fig. 1.

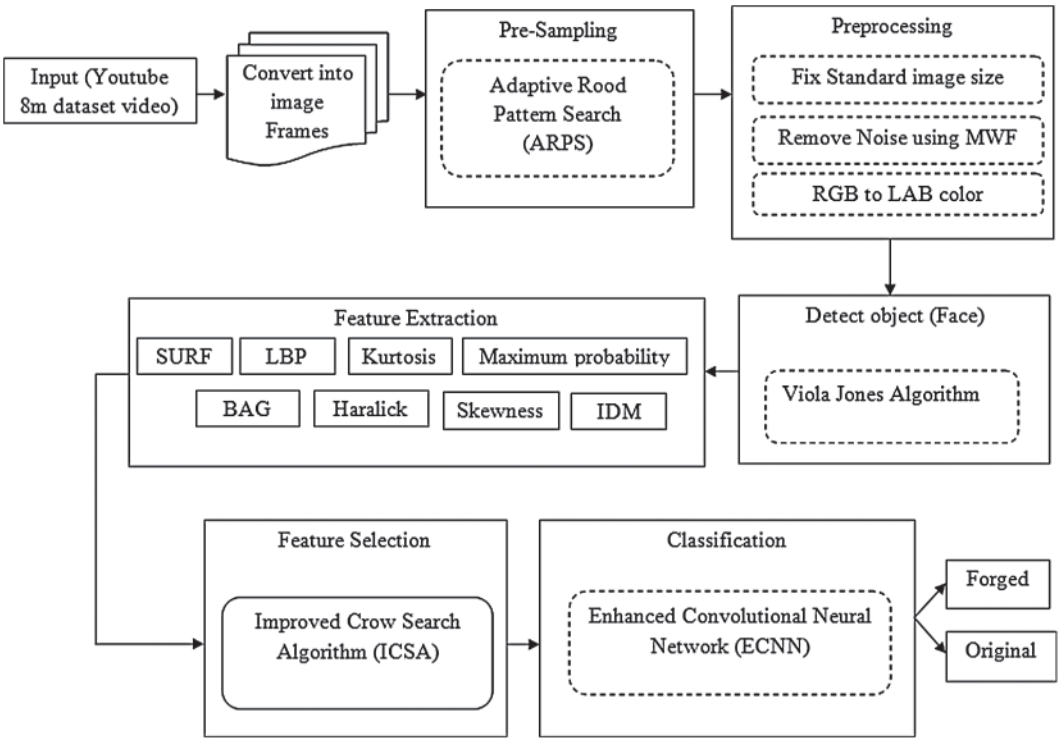

Figure 1: Workflow of the proposed detection system

3.2 Data Collection

The proposed system has collected the data from YouTube for evaluation of performance. The presented videos having a resolution bigger than 480p, which was tagged with "face," "newscaster," or "news program" in the youtube8m data-set was taken. YouTube-8M is an enormous scale marked video dataset that comprises of a large number of YouTube video IDs, with excellent machine-produced comments from various jargon of 3,800+ visual substances. It accompanies recomputed various media highlights from billions of edges and sound fragments, intended to fit on a solitary hard circle. This makes it conceivable to prepare a robust benchmark model on this dataset in under a day on a single GPU Simultaneously, the dataset's scale and decent variety can empower profound investigation of complex, broad media models that can take a long time to prepare even in a circulated manner. Fig. 2 shows the video frame original, and Fig. 3 shows the tampered video frame.

3.3 Convert Videos into Frames

The inputted video is originally converted into frames that are mathematically written as Eq. (1),

$$I = \{f_1, f_2, f_3, \ldots, f_n\} \tag{1}$$

where I denotes the video frameset and f_n represents the 'n'-number of frames.

Figure 2: Video frame original

Figure 3: Video frame forgery

3.4 Pre-Sampling

At this stage, the use of the ARPS reduces the number of redundant frames (useless frames). Symmetrical patterns of search refer to the shape of "rood" or "cross." There will be '2' distinct search phases called the Adaptive Rood Pattern (ARP) besides the Unity Rood Pattern (URP) that differ mainly in search points separation. Of course, the ARPS are iterative; the ARP phase is used once, until the search converges, the lightweight URP search is implemented iteratively. Fig. 4 shows the search sequence of ARPS algorithm.

3.5 Preprocessing

The resampled frames are preprocessed. The preprocessing stage involves choosing the standard image size, removal of noise and 'RGB' to 'LAB' color spaces conversion which is explained as follows.

(a) Standard Image Size

In this phase, the image is set with a fixed size. If the system gets a different size of images for performing the task, then it may produce unwanted or error results. To avoid such drawbacks, the input images are initially fixed at standard size.

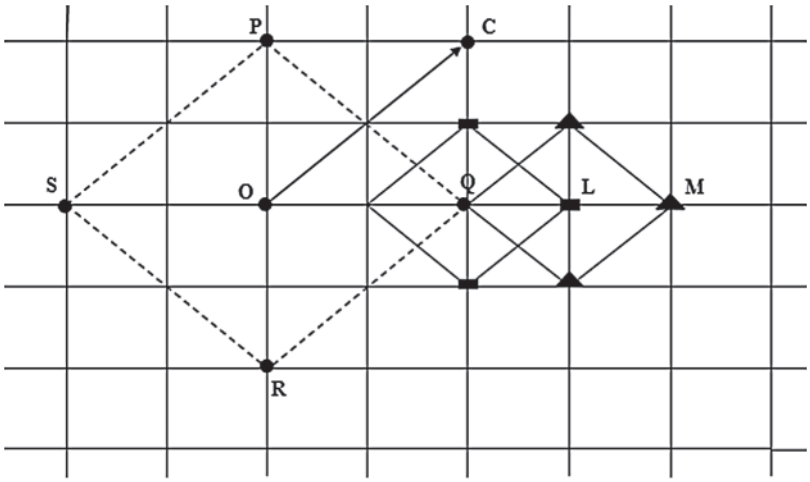

Figure 4: Search sequence for ARPS algorithm

(b) Removal of Noise

In this stage, the salt and pepper noise has been eliminated using Modified Weiner Filter (MWF), which produces a better result than the ordinary Weiner Filter (WF). It is also more effective in preserving the edges. It is a Pixel wise linear filter formed by evaluating the local mean and variance about each pixel. The image I_p value at a point $(u,\ v)$ is,

$$I_E = I_p(u,v) = \eta + \frac{\left(\sigma^2 - r^2\right)}{\sigma^2}\left(I_i(u,v) - \eta\right) \tag{2}$$

Eq. (2) indicates that η is the area's mean that is under consideration, σ^2 implies the variance, r^2 is the noise variance, I_i is the image under consideration for noise removal, and I_E denotes the enhanced image that is utilized for further steps.

(c) RGB to LAB Space Conversion

After removing salt and peppers' noise from images then it transformed into $L^*a^*b^*$ color space. Therefore, the transformation is separated by two steps: (i) RGB to XYZ (tristimulus values (i.e., coordinate values of the RGB image)), and (ii) XYZ to $L^*a^*b^*$.

(i) RGB to XYZ

Assume that $r,\ g,\ b$ are three channels of pixels, and the gamut of values is [0, 255]. The conversion formula is as Eq. (3):

$$\begin{cases} R = gamma\left(\dfrac{r}{255}\right) \\ G = gamma\left(\dfrac{g}{255}\right) \\ B = gamma\left(\dfrac{b}{255}\right) \end{cases} \tag{3}$$

The XYZ conversion is,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = A \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4}$$

In Eq. (4) A denotes the 3×3 matrix value of the gamma function of the RGB value.

(ii) XYZ to $L^*a^*b^*$

In this stage, the obtained XYZ components are converted into three components $L^*a^*b^*$ which is expressed as follows:

$$L^*a^*b^* = \begin{Bmatrix} L^* = 116 c_f \left(\frac{Y}{Y_n}\right) - 16 \\ a^* = 500 \left[c_f \left(\frac{X}{X_n}\right) - c_f \left(\frac{Z}{Z_n}\right) \right] \\ b^* = 200 \left[c_f \left(\frac{X}{X_n}\right) - c_f \left(\frac{Z}{Z_n}\right) \right] \end{Bmatrix} \tag{5}$$

In Eq. (5) X_n, Y_n and Z_n denotes the tristimulus values measure light intensity based on the three primary color values (RGB), typically represented by X, Y, and Z coordinates as well as the c_f, represents the calibration function which is stated as follows,

$$c_f(t) = \begin{cases} t^{\frac{1}{3}} & if \;\; t > \eta^3 \\ \frac{t}{3\eta^2} + \frac{4}{29} & otherwise \end{cases} \tag{6}$$

Eq. (6) indicates that η value is $\frac{6}{29}$. Finally, the $L^*a^*b^*$ converted image is obtained.

3.6 Detect Objects (Face) Using Viola–Jones Algorithm

The pre-processed picture obtained from the color image of the LAB is defined by the object. In the image provided, Viola and Jones offer pace and effective ways to recognize a face. This algorithm includes the following steps in a significant way.

3.6.1 Haar Like Features

This was a rectangular digital image part that originates from its resemblance to Haar-wavelets. The two-component value of the rectangle is the variance between the sum of the pixels between the rectangular regions of '2'. The balance is determined by a three-rectangle function in the middle of the two outer rectangles that decreased from the rectangle center number. A four-rectangle component essentially tests the diagonal pairs of the variance between rectangles. The significance of rectangular features is measured as

$$V_{RF} = \left[\sum P_{A(black)} - \sum P_{A(white)} \right] \tag{7}$$

In Eq. (7) V_{RF} denotes that features value and the pixels of black and white area are denotes as $P_{A(black)}$ and $P_{A(white)}$ respectively.

3.6.2 Feature Selection and Analysis

The Viola–Jones technique used the modification of the Ada Boost algorithm created by Freund and Schapire to construct an efficacy classifier by tabbing a minimum number of critical components. The training data will include photographs for the effects of preeminence through the spectrum of lighting conditions and facial properties.

3.6.3 Integral Image

An innovative representation of images called an integral image allows evaluation of very fast features. The detection system is not directly performing with image intensities here. This method of object detection categorizes images by taking into account the meaning of basic features. The integral image is determined from a single pixel image that requires certain operations. All of these hair-like components are measured in stable time at [any] location or scale.

The integral position figure $(i,\ j)$ includes the sum of the pixels above and the left part of '$i,\ j$' including,

$$I_i\,(i,j) = \sum_{x'<x',y'<y'} I_E\,(i,j) \tag{8}$$

In Eq. (8) $I_i\,(i,\ j)$ is the main image and $I_E\,(i,\ j)$ is the real image intensity.

3.6.4 Ada Boost Algorithm

The object detection system provides an Adaboost learning algorithm for both tabbing the best feature and classifying training that uses it. This algorithm therefore generates a strong classifier as a weighted linear combination of simple weak classifiers.

$$h\,(x) = sign\left(\sum_{j=1}^{N} \beta_j h_j\,(x)\right) \tag{9}$$

Each weak classifier has a threshold function of feature F_j. Eq. (9) denotes that weak classifier of $h_j\,(x)$ having a feature F_j and limitation of threshold where indicated by θ_j and a parity that denotes the in-equality sign path:

$$h_j\,(x) = \begin{cases} -T_j & if\ F_j < \theta_j \\ T_j & if\ otherwise \end{cases} \tag{10}$$

Eq. (10) indicates that threshold value of $\theta_j T_j$ to ensure training and co-efficient, x and β_j where includes a 24-by-24 image sub-window.

3.6.5 Cascade Architecture

The input window evaluation is performed on the cascade's initial classifier, and if that classifier arrives wrong, the window's measurement is completed, and the detector arrives correct. If the classifier reappears true, however, the window will be passed to the following cascade classifier. Since most of the image windows may not look like faces, most are quickly rejected as non-faces.

3.7 Feature Extraction

In this phase, the SURF, LBP, Kurtosis, Maximum probability and BAG features are extorted as of the pre-processed image which is explained as follows:

(a) Speeded Up Robust Feature (SURF) Feature

This method uses a BLOB (Binary Large Objects) Hessian matrix detector to address the stains. This method used wave-let responses in horizontal along with vertical directions by using appropriate Gaussian weights for the assignment of feature definition and orientation. A neighbor about pivotal point is identified and divided into sub-regions, and then the wave-let responses are considered and indicated for each sub-region to obtain SURF. Eq. (11) shows that descriptor vector '$d(y)$' for every sub-region.

$$SURF = d(y) = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right) \tag{11}$$

(b) Local Binary Pattern

The LBP is a texture centered feature that has extensive applications in image classification. The LBP feature is provided as,

$$LBP = \sum_{s=1}^{n} 2^s * S_f(I_N - I_C) \tag{12}$$

In Eq. (12) I_N signifies the neighboring pixel on a square window, I_C implies the center pixel in the square window, 's' signifies the number of neighboring pixels around a center pixel. 'S_f' signifies specific function and $(I_N - I_C)$ is marked as the thresholds value and it is estimated by Eq. (13).

$$S_f(I_N - I_C) = \begin{cases} 1, & if\ I_N - I_C \geq 0 \\ 0, & if\ I_N - I_C < 0 \end{cases} \tag{13}$$

(c) Kurtosis

The shape of an arbitrary variable's probability distribution is delineated using the parameter called Kurtosis. For the arbitrary variable 'X', the Kurtosis is denoted as, $K_{urt}(X)$ and it is stated as,

$$K_{urt}(X) = \left(\frac{1}{m \times n}\right) \frac{\sum (f(x,y) - M)^4}{SD^4} \tag{14}$$

In Eq. (14) SD denotes the standard deviation and M represents the mean value.

(d) Maximum Probability

Maximum probability is merely the most significant entry in the matrix as well as equivalents to the most robust response calculated by Eq. (15).

$$Maximum\ probability = Max\,\|f(x,y)\| \tag{15}$$

(e) Block Artificial Grid (BAG)

It has some visually horizontal or vertical breaks on the image. This termed BAG emerges at the border of every pixel block. This can well be utilized for resolving whether the image is

changed or not. If the entire BAGs are extracted as of a specified image, regions with BAGs with-in the block border are deliberated as forged areas. Final BAG is attained by adding two elements (image's vertical and horizontal edge) in

$$BAG = b(x,y) = b_h(x,y) + b_v(x,y) \tag{16}$$

In Eq. (16) b_v and b_h denotes the image's vertical and horizontal edge.

3.8 Feature Selection

The obligatory features are selected as of the extorted features for avoiding the execution time. The necessary features are selected using ICSA which is enlightened as follows.

3.8.1 Improved Crow Search Algorithm

The crows' intelligence activities are enthusiastic about the Crow search algorithm (CSA). The CSA has developed its potential to achieve the optimal solution for specific configurations of search spaces. Nevertheless, due to the unproductive discovery of its quest policy, its convergence is not certain. Under this situation, when faced with higher multimodal formulations, the search approach faces great challenges. This proposed method uses the ICSA to solve these difficulties. The improvement is accomplished by adding Levy's fight for random movement performance. In Lévy flights, a heavy-tailed distribution of probabilities, called the Lévy distribution, controls the phase scale. The search space is essentially discovered by the Lévy Flights relative to the uniform random distribution.

The evolutionary process of the CSA imitates the activities carried out using crows to hide and recover the additional food. As an algorithm centered on population, the size of the flock is confirmed by 'N' individuals (crows) that are of n-dimensional with 'n' as the problem dimension. The position '$W_{i,k}$' of the crow g in the specific iteration k is illustrated in Eq. (17) and signifies a probable solution for the issue:

$$W_{i,k} = \left[w_{i,k}^1, w_{i,k}^2, \ldots, w_{i,k}^n\right], \quad i = 1,2,\ldots,N; \quad k = 1,2,\ldots,\max\ Iter \tag{17}$$

In Eq. (17) max *Iter* is the maximum of iterations. Each (individual) crow is supposed to be capable of memorizing the better visited location $R_{i,k}$ to hide food until the present iteration in Eq. (18).

$$R_{g,k} = \left[r_{g,k}^1, r_{g,k}^2, \ldots, r_{g,k}^n\right] \tag{18}$$

The position of everyone is modified as per the two behaviors such as Pursuit and evasion.

Pursuit: The crow 'h' has follows the crow 'g' along with specific reason to find a hidden place. Moreover crow 'g' does not note the other crow's presence, as a result, the crow's h justification is attained.

The sort of behavior regarded by every crow g is determined y the Alertness probability (AP). So, a random value a_g uniformly distributed between '0' and '1' is sampled. Levy flights fundamentally give a random walk, the random steps of which are drawn as of a Levy distribution for significant steps:

$$a_g = Levy \sim t^{-\lambda}, \quad (1 < \lambda \leq 3) \tag{19}$$

Eq. (19) denotes that a_g is greater or equal than AP, and the behavior one is applied. Otherwise, situation two is chosen. This operation can be summarized in the subsequent model:

$$W_{g,k+1} = \begin{cases} W_{g,k} + a_g \cdot fl_{g,k} \cdot \left(R_{h,k} - W_{g,k}\right) & a_i \geq AP \\ random & otherwise \end{cases} \tag{20}$$

In Eq. (20) flight length $fl_{g,k}$ parameter indicates the magnitude of movement from crow $W_{g,k}$ towards the best position $W_{h,k}$ of the crow h, a_g signifies a random number having a uniform distribution on the gamut [0, 1]. The proposed ICSA algorithm's Pseudocode is exhibited in Fig. 5.

Input : Extracted Features
Output : Selected Features

Begin
Randomly initialize the position of a flock of N crows in the search space
Evaluate the position of the crows
Initialize the memory of each crow
While $Iter < Iter_{max}$
for i=1 ; N
Randomly choose one of the crows to follow
Define AP
if $a_g \geq AP^{h,iter}$
$W_{g,k+1} = W_{g,k} + a_g.fl_{g,k}.(R_{h,k} - W_{g,k})$
else
$W_{g,k+1}$ = a random position of search space
end if
end for
Check the feasibility of new positions
Evaluate the new position of the crows
Update the memory of crows
end while

Figure 5: Pseudocode for proposed ICSA algorithm

3.9 Classification Utilizing ECNN

This section carried out the classification. The selected features are inputted to the ECNN. ECNN contains, along with local connectivity, a few conceptions called parameter sharing. Sharing parameters is the sharing of weights on a specific feature map (FM) through each neuron. Local connectivity is the concept of each neural connected only to a sub-set of the input image; this helps to reduce the total parameters of the system as a whole and makes the computation more efficient. A Visual patterns are easily detected by spotting ECNN's shared weight. In shared weight property, the ECNN model uses replicated filters that have matching weight vectors as well as have local connectivity. Taking the well-known LeNet-5, it comprises three sorts of layers, explicitly convolutional, pooling, as well as fully-connected layers. The convolutional layer intends to learn feature representations of the inputs. The architecture diagram of the ECNN is exhibited in Fig. 6.

In decipher convolution by setting up one vector, striding along with it the other vector, and processing a spot item for each walk. That spot element provides one number in the yield vector that is a section. Based on ECNNs, convolution layer is commonly made out of 3 stages. The

principal arrange includes learnable channels, parameterized each playing out a convolution in parallel. This convolution activity can be changed from the above definition in various manners, for example, the amount to walk before figuring another dab item. The subsequent stage includes a component savvy non-linearity like a completely associated layer. At long last, the third stage is called pooling. Pooling is a technique for down sampling the yield vector of the subsequent stage. One approach to do this is called max-pooling, in which the maximal component in a characterized segment of the yield is taken to speak to the whole area. To abridge, a convolutional layer attempts to discover nearby examples in the info. Each channel in the main stage is found out during preparing, so that is task explicit. As it were, the ECNN endeavors to locate the most pertinent examples that help decide how to achieve the given assignment. An identical perspective about ECNNs is by envisioning a convolutional layer just like a wholly associated layer with a limitlessly solid earlier that says loads are shared crosswise over info information passages, and a dominant part of them are zero.

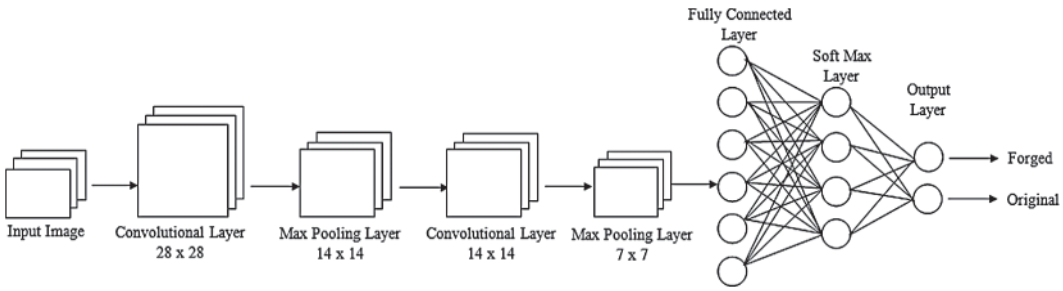

Figure 6: Architecture of ECNN

Specifically, each FM neuron is connected to the adjacent neuronal region on the other side. This neighborhood is referred to as the receptive region of the neuron on the former ground. It is possible to achieve the new FM by initially converting the input with a trained kernel and then applying an element-wise nonlinear activation function on the converted output. Using several different kernels, the entire FM is reached. Mathematically, the feature value at a location (a, b) in the kth FM of lth the layer $q^l_{a,b,k}$ is computed by:

$$q^l_{a,b,k} = w^l_k \, x^l_{a,b} + b^l_k \tag{21}$$

In Eq. (21) w^l_k and b^l_k signify the weight factor as well as bias term of the kth filter of the lth layer correspondingly, and $x^l_{a,b}$ implies the input patch grounded location (a, b) of the lth layer. Note that the kernel w^l_k that generates the feature map $q^l_{a,b,k}$ is shared. Such a weight sharing mechanism encompasses numerous pros; for instance, it can lessen the model intricacy and make the network easy to train. The activation function commences non-linearities to CNN that are enviable for multi-layer networks to detect non-linear features. Let $n(\cdot)$ implies the nonlinear activation function. The activation value $(n)^l_{a,b,k}$ of the convolution feature $q^l_{a,b,k}$ can well be calculated as shown in Eq. (22).

$$(n)^l_{a,b,k} = n\left(q^l_{a,b,k}\right) \tag{22}$$

Classic activation functions are tan, sigmoid, along with ReLU. The pooling layer intends to attain shift-invariance by lessening the FM resolution. It is placed usually betwixt '2' convolutional layers. Every FM of a pooling layer is joined to its equivalent FM of the former convolutional layer. Implying the pooling function as $p(\cdot)$ aimed at all feature map $(n)^l_{a,b,k}$:

$$y^l_{a,b,k} = p\left((n)^l_{m,n,k}\right), \forall (m,n) \in R_{ab} \tag{23}$$

In Eq. (23) R_{ab} implies a local neighborhood about location (a, b). The classic pooling functions are average pooling along with max pooling.

After numerous convolutional as well as pooling layers, there might be more than one fully-connected layer that intends to do higher-level reasoning. They take every neuron on the former layer and join them to each neuron of the present-layer to make universal semantic information. Take into account that the fully-connected layer is always not essential as it can be swapped by a [1 × 1] convolution layer.

The last CNN layer is the output layer. Aimed at classification, the softmax operator is generally utilized. Let 'θ' implies every CNN parameters (for instance, the weight vectors as well as bias terms). The optimal parameters intended for a specific task can well be attained by lessening a suitable loss function stated in that task. This proposed method has N desired input-output relations $\left\{\left(x^{(n)}, y^{(n)}\right); n \in [1, \ldots, N]\right\}$, where $x^{(n)}$ implies the nth input data $y^{(n)}$ signifies its corresponding target label and $o^{(n)}$ is the CNN output. The loss of CNN can well be computed as follows:

$$L_f = \frac{1}{N}\sum_{n=1}^{N} l\left(\theta, y^{(n)}, o^{(n)}\right) \tag{24}$$

In Eq. (24) L_f denotes the loss function. Training CNN is an issue of universal optimization. Ensure whether the L_f is maximum or minimum, on the off-chance that it is minimum, then train this model. If the L_f is maximum, then the weight parameter is updated using "Adam Optimizer algorithm" again calculate the L_f Adam optimizer is explained as follows:

Adam is primarily derived as of the adaptive moment assessment. Mainly, this algorithm track the first '2' uncentered moments c_t and υ_t of the gradient of the objective function at every time step:

$$r_t = \nabla_{L_f} f_t\left(\left(L_f\right)_{t-1}\right) \tag{25}$$

In Eq. (25) $f_t\left(\left(L_f\right)_{t-1}\right)$ denotes the evaluation of the fixed loss function on the particular batch of data occurring at timestep $t-1$, r_t indicates the gradient vector. Initially, c_t and υ_t value is assigned as 0. Then the uncentered moments are estimated for each weight value.

$$c_t = \beta_1 . c_{t-1} + (1-\beta_1) . r_t \tag{26}$$

$$\upsilon_t = \beta_2 . \upsilon_{t-1} + (1-\beta_2) . (r_t)^2 \tag{27}$$

In Eqs. (26) and (27) $\beta_1\beta_2$ and denotes the exponential decay rates intended for the moment estimates. After calculated $c_t\upsilon_t$ and for each weight value c_t and υ_t will be modified by the power of the current frame index value as (Eqs. (28) and (29)).

$$\hat{c}_t = c_t / \left(1 - \beta_1^t\right) \tag{28}$$

$$\hat{\upsilon}_t = \upsilon_t / \left(1 - \beta_2^t\right) \tag{29}$$

Finally, each weight value would be updated as to its previous values $\hat{c}_t$ and $\hat{\upsilon}_t$.

$$\theta_t = \theta_{t-1} - \eta \left(\hat{c}_t / \sqrt{\hat{\upsilon}_t} + \varepsilon\right) \tag{30}$$

In Eq. (30) θ_t denotes the updated weight value, θ_{t-1} indicates the previous weight value, η represents the learning rate, ε denotes the tolerance parameter, which is to prevent the division from zero error. For using this way, the weight parameters are updated and to diminish the L_f. By using this ECNN, the image frame is detected as the forged image frame or the original image frame.

4 Experimantal Result and Discussion

This Section presents the experimental results of the proposed ECNN for the forgery detection technique in the image frames of the videos. Simulation of the proposed ECNN carried out by the MATLAB with machine configuration as Processor: Itel Core i7, OS: Windows 7, CPU speed: 3.20 GHz, RAM: 8 GB. All test successions were picked to ensure high spatial and worldly circulations. A few recordings have a higher spatial 247 action (surface), while others have higher swift action (development). The sound substance we consider to select 248 successions that had discourse, music, and encompassing sound. This assorted variety is a significant necessity while picking a video 249 database for testing altering assaults. Tab. 1 shows the common video tampering clues used in this research work.

Table 1: Common video tampering clues used in this work

Tampering clues	Copy-move	Cut-paste	Delete-fill	Localization
Composition	✗	✓	✓	✓
Cropping	✓	✓	✓	✓
Flipping	✓	✗	✓	✓
Salt and Pepper	✓	✓	✗	✓
Scaling	✗	✓	✓	✓
Rotating	✓	✓	✓	✓

4.1 Performance Analysis

Here, the proposed method's performance is done by determining a discrete number of performance parameters. The basic parameters that are calculated are 'true positive' (*TP*), 'true negative' (*TN*), 'false positive' (*FP*) and 'false negative' (*FN*) values. The proposed scheme is contrasted and evaluated centered upon the performance metrics, for instance, accuracy, specificity, and sensitivity, and F-score.

Accuracy: The accuracy refers to the closeness to the actual performance, and it is measured as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Specificity: The specificity refers to the actual negative case forged video frame got predicated correctly.

$$Specificity = \frac{TN}{FP + TN}$$

Sensitivity: Sensitivity defines the actual positive cases forged video frame, which got predicted correctly, and it is calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN}$$

F-Score: The F score, also called the F1 score or F measure, is a measure of a test's accuracy.

$$F-Score = 2 * \frac{Sensitivity * Specificity}{Sensitivity + Specificity}$$

4.2 Comparative Analysis

The proposed ECNN classifier is compared with the existing methods such as SVM and K-Nearest Neighbor (KNN), Fruitfly optimization algorithm-support vector-NN (FOA-SVNN) and Neural Network (NN) and each works is discussed in detail as follows:

SVM: SVM is a binary classifier. It attempts to find a hyperplane that can separate two class of data by the most significant margin.

K-Nearest Neighbor (KNN): The k-NN method accepts that an unclassified thing can be characterized by taking a looking at k of its effectively ordered, closest neighbors and discovering which class the most significant number of them fall into.

Fruitfly optimization algorithm-support vector-NN (FOA-SVNN) can adaptively determine the two critical hyper-parameters for SVM.

Neural Network (NN): A neural network can be used for many different tasks. One of these tasks is classification.

The performance analysis value of the proposed forgery detection system using ECNN and existing techniques, for instance, SVM and K-Nearest Neighbor (KNN), Fruitfly optimization algorithm-support vector-NN (FOA-SVNN) and Neural Network (NN) for different metrics comparison are shown in below Tab. 2.

The preceding Tab. 2 displays the comparison of the proposed ECNN forgery detection technique with the existent techniques in respects of accuracy, specificity, sensitivity. From Tab. 2, it is clear that the existing NN method has provided the bad performance than the other existent methods. Next, existing KNN is a lot better than the NN in the forgery detection system. ECNNs are generally excellent feature extractors. This implies you can remove helpful characteristics from a previously prepared ECNN with its prepared loads by nourishing your information on each level and tune the ECNN a piece for the particular errand. ECNNs are exceptionally useful in such a task contrasted with NNs. Another favorable position of this pre-preparing is that we

abstain from preparing of ECNN and spare memory, time. The main thing you need to prepare is the classifier toward the end for your labels. From overall observation based on the table value, it proves that the proposed ECNN forgery detection system provides better performance than the other existing methods.

Table 2: Demonstrate the performance of the proposed ECNN with the existing FOA-SVNN, SVM, KNN and NN

Performance metrics	ECNN	FOA-SVNN	SVM	KNN	NN
Accuracy	97.21	94	87.04	79.79	87.84
Specificity	98.56	95.83	94	94.67	94
Sensitivity	95.67	94	67.01	55.88	68.97

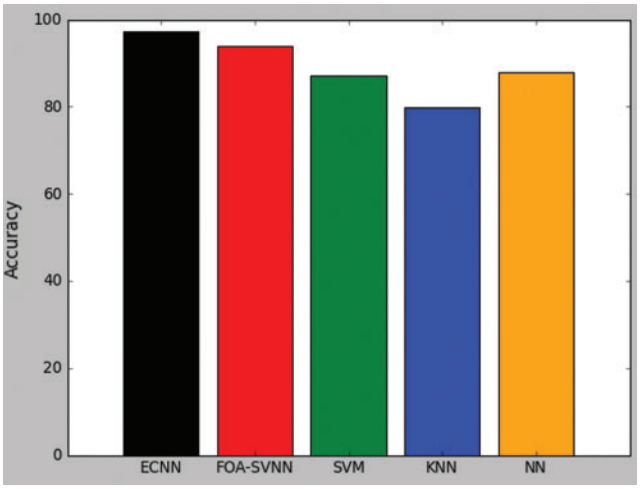

Figure 7: Accuracy performance comparison for ECNN with existing methods

Fig. 7 shows the performance of the proposed ECNN with the existing SVM, FOA-SVNN, KNN, and NN based on accuracy metric. The proposed ECNN achieves 97.21 accuracy, but the existing FOA-SVNN achieves 94 accuracy, which is 32.1 lesser than the proposed technique. Also, the other existing methods, namely SVM, KNN, as well as NN, have 94, 94.67, and 94 accuracy. This discussion exhibits that the ECNN encompass better performance when weighted against the existent methods.

Fig. 8 exhibits the comparison graph for the proposed ECNN with the existing techniques based on specificity measure. The proposed method's specificity is 98.56. The existing systems FOA-SVNN, SVM, KNN, and NN, have 95.83, 94, 94.67, and 94, respectively. Hence it proves that the specificity value is high for the proposed work than the existing systems. k-NN is straightforward and requires tuning only one hyperparameter (the estimation of k), while neural network preparing includes numerous hyperparameters controlling the size and structure of the system and the enhancement methodology, so neural network provides high specificity than KNN algorithm. The excellence of the ECNN lies in utilizing the preparing power accessible to let the model gain proficiency with all the bit qualities to accomplish the last focus of the model,

which could be a necessary arrangement. At the point when you take a look at the halfway pictures during derivation, you will perceive how the various hubs are distinguishing various edges, shading, and so on, in the first convolution layers only like PC vision channels. Be that as it may, the expansion of this separating prompts distinguishing progressively complex examples in resulting Convolution layers. The proposed ECNN performed better with the value for specificity as 98.56.

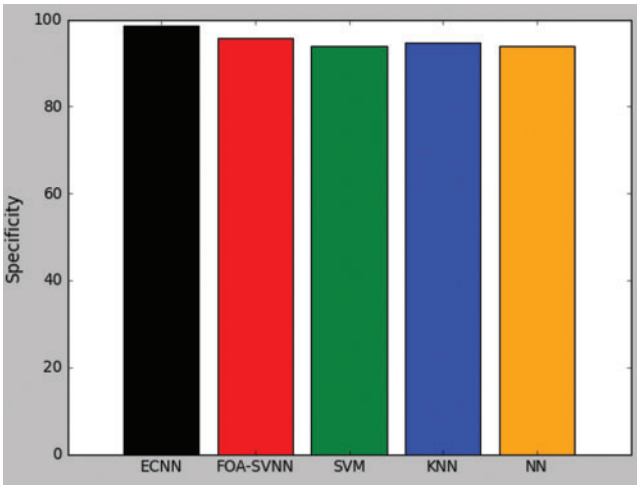

Figure 8: Demonstrate the performance of the proposed ECNN with existing detection methods in terms of specificity

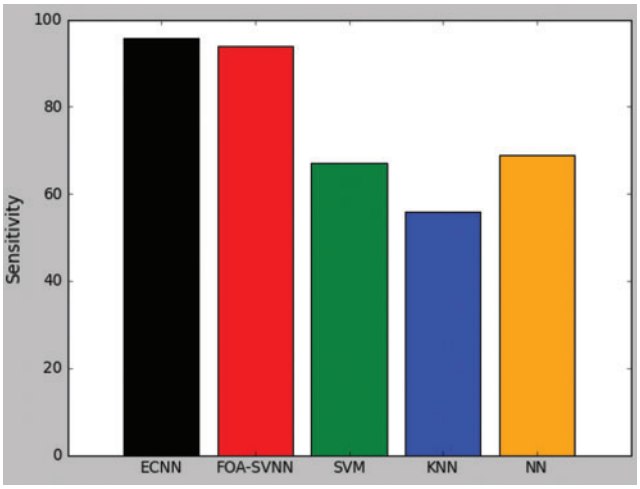

Figure 9: Sensitivity comparison graph for the proposed ECNN technique with existing techniques

Fig. 9 compared the performance of the ECNN with the existing forgery detection techniques based on the sensitivity metric. ECNN achieves 95.67 sensitivity, existing FOA-SVNN obtain 94 sensitivity, SVM has 67.01, KNN achieves 55.88, and the NN has 68.97. It exhibits that the ECNN system has offered better performance when weighted against the existent methods.

Tab. 3 shows the F-score value of proposed ECNN with the existing FOA-SVNN, SVM, KNN, and NN.

Table 3: Demonstrate the F-score of the proposed ECNN with the existing FOA-SVNN, SVM, KNN and NN

Performance metrics	ECNN	FOA-SVNN	SVM	KNN	NN
F-Score	97.09349946	94.90617921	78.24284206	70.27777615	79.56286433

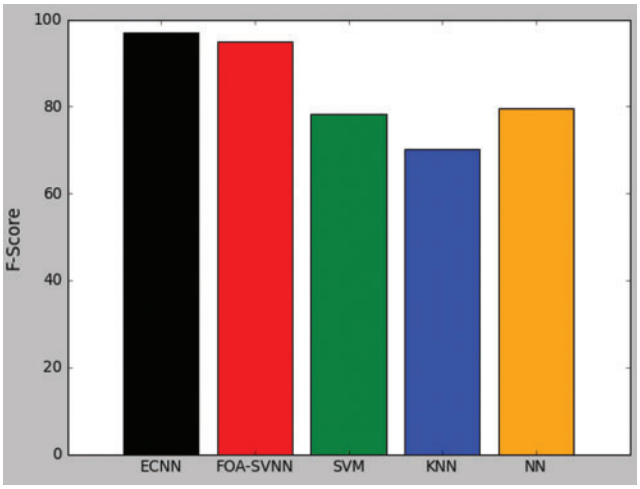

Figure 10: F-score comparison graph for the proposed ECNN technique with existing techniques

F Score is the weighted normal of sensitivity and specificity. Like this, this score considers both bogus positives and false negatives. Naturally, it is not as straightforward as exactness. However, F is generally more valuable than precision, particularly on the off chance that you have a lopsided class circulation. From Fig. 10, we can understand that ECNN provides a better F score than the Existing method.

5 Conclusion

Nowadays, many multimedia tools and applications are available that utilized to edit or temper medical files. Most of current detection methods are not performed well in the accuracy level. Here, the forgery detection image frames of the videos are done using ECNN proves the performance. The input video is transmuted into frames, next the frames are pre-sampled. Then, perform the preprocessing process for ameliorating the image frames. After that, select the features from the pre-processed images. Then, select the necessary features as of the extorted features using the ICSA algorithm, in the final stage, the image is classified centered on the preferred features using ECNN. The proposed classification technique's performance is weighed against that of the existent techniques.The proposed ECNN performance is compared with the existent FOA-SVNN, SVM, KNN, as well as NN in terms of sensitivity, accuracy, and also specificity. The Experimental outcomes exhibited that the proposed ECNN classifies the objects more accurately than the existent methods. The proposed method has achieved high F-Score of 97.09 compare to

other existing work. In the future, this proposed work can extend by using different optimization techniques to achieve more accuracy.

Funding Statement: This work does not receive any funding or financial support from any agencies.

Conflicts of Interest: Authors declare that there is no conflict of interest.

References

1. Birajdar, G. K., Mankar, V. H. (2013). Digital image forgery detection using passive techniques: a survey. *Digital Investigation, 10(3),* 226–245. DOI 10.1016/j.diin.2013.04.007.
2. Snigdha Mankar, K., AjayGurjar, A. (2015). Image forgery types and their detection: a review. *International Journal of Advanced Research in Computer Science and Software Engineering, 5(4),* 60–68.
3. Abdalla, Y. E., Iqbal, M. T., Shehata, M. (2018). Fusion approaches system of copy-move forgery detection. *American Journal of Computer Science and Engineering Survey, 6(1),* 1–12.
4. Kaur, A., Vats, I. (2017). Authentication based image forgery detection using optimized features in JPEG images. *International Journal of Advanced Research in Computer Science, 8(7),* 616–621. DOI 10.26483/ijarcs.v8i7.4316.
5. Sharma, V., Jha, S., Bharti, R. K. (2016). Image forgery and it's detection technique: a review. *International Research Journal of Engineering and Technology, 3(3),* 756–762.
6. Zhang, Y., Zhao, C., Pi, Y., Li, S. (2012). Revealing image splicing forgery using local binary patterns of DCT coefficients. *Communications, Signal Processing and Systems,* USA, 181–189.
7. Lad, M., Patel, N. (2016). Passive digital image forgery detection techniques and implementation. *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, 4(5),* 417–424.
8. Bianchi, T., Piva, A. (2012). Image forgery localization via block-grained analysis of JPEG artifacts. *IEEE Transactions on Information Forensics and Security, 7(3),* 1003–1017. DOI 10.1109/TIFS.2012.2187516.
9. Kang, X., Stamm, M. C., Peng, A., Liu, K. J. R. (2013). Robust median filtering forensics using an autoregressive model. *IEEE Transactions on Information Forensics and Security, 8(9),* 1456–1468. DOI 10.1109/TIFS.2013.2273394.
10. Muhammad, G., Al-Hammadi, M. H., Hussain, M., Bebis, G. (2014). Image forgery detection using steerable pyramid transform and local binary pattern. *Machine Vision and Applications, 25(4),* 985–995. DOI 10.1007/s00138-013-0547-4.
11. Li, J., Li, X., Yang, B., Sun, X. (2014). Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security, 10(3),* 507–518.
12. Prakash, C. S., Kumar, A., Maheshkar, S., Maheshkar, V. (2018). An integrated method of copy-move and splicing for image forgery detection. *Multimedia Tools and Applications, 77(20),* 26939–26963.
13. Hu, W. C., Chen, W. H., Huang, D. Y., Yang, C. Y. (2016). Effective image forgery detection of tampered foreground or background image based on image watermarking and alpha mattes. *Multimedia Tools and Applications, 75(6),* 3495–3516. DOI 10.1007/s11042-015-2449-0.
14. Bhartiya, G., Jalal, A. S. (2017). Forgery detection using feature-clustering in recompressed JPEG images. *Multimedia Tools and Applications, 76(20),* 20799–20814. DOI 10.1007/s11042-016-3964-3.
15. Mahmood, T., Mehmood, Z., Shah, M., Saba, T. (2018). A robust technique for copy-move forgery detection and localization in digital images via stationary wavelet and discrete cosine transform. *Journal of Visual Communication and Image Representation, 53,* 202–214. DOI 10.1016/j.jvcir.2018.03.015.
16. Elsharkawy, Z. F., Abdelwahab, S. A. S., Abd El-Samie, F. E., Dessouky, M., Elaraby, S. (2019). New and efficient blind detection algorithm for digital image forgery using homomorphic image processing. *Multimedia Tools and Applications, 78(15),* 21585–21611. DOI 10.1007/s11042-019-7206-3.

17. Oommen, R. S., Jayamohan, M., Sruthy, S. (2016). Using fractal dimension and singular values for image forgery detection and localization. *Procedia Technology, 24,* 1452–1459. DOI 10.1016/j.protcy.2016.05.176.
18. Youseph, S., Cherian, R. R. (2015). Pixel and edge based illuminant color estimation for image forgery detection. *Procedia Computer Science, 46,* 1635–1642. DOI 10.1016/j.procs.2015.02.099.
19. Pun, C. M., Yuan, X. C., Bi, X. L. (2015). Image forgery detection using adaptive over segmentation and feature point matching. *IEEE Transactions on Information Forensics and Security, 10(8),* 1705–1716. DOI 10.1109/TIFS.2015.2423261.
20. Zhong, J., Gan, Y., Young, J., Huang, L., Lin, P. (2017). A new block-based method for copy move forgery detection under image geometric transforms. *Multimedia Tools and Applications, 76(13),* 14887–14903. DOI 10.1007/s11042-016-4201-9.
21. Barani, M. J., Valandar, M. Y., Ayubi, P. (2019). A new digital image tamper detection algorithm based on integer wavelet transform and secured by encrypted authentication sequence with 3D quantum map. *Optik, 187,* 205–222. DOI 10.1016/j.ijleo.2019.04.074.
22. Johnston, P., Elyan, E. (2019). A review of digital video tampering: from simple editing to full synthesis. *Digital Investigation, 29,* 67–81. DOI 10.1016/j.diin.2019.03.006.
23. Hong, J. H., Yang, Y., Oh, B. T. (2019). Detection of frame deletion in HEVC-coded video in the compressed domain. *Digital Investigation, 30,* 23–31. DOI 10.1016/j.diin.2019.06.002.
24. Uliyan, D. M., Sadeghi, S., Jalab, H. A. (2019). Anti-spoofing method for fingerprint recognition using patch based deep learning machine. *Engineering Science and Technology, 23(2),* 264–273.
25. Bi, X., Pun, C. M. (2018). Fast copy-move forgery detection using local bidirectional coherency error refinement. *Pattern Recognition, 81,* 161–175. DOI 10.1016/j.patcog.2018.03.028.
26. Velliangiri, S. (2020). An enhanced multimedia video surveillance security using wavelet encryption framework. *Journal of Mobile Multimedia, 15(3),* 239–254.