IET Image Processing

Research Article

IET Journals
The Institution of Engineering and Technology

Illumination-based texture descriptor and fruitfly support vector neural network for image forgery detection in face images

ISSN 1751-9659
Received on 21st October 2017
Revised 22nd February 2018
Accepted on 16th March 2018
E-First on 30th April 2018
doi: 10.1049/iet-ipr.2017.1120
www.ietdl.org

Rajan Cristin[1] ✉, John Patrick Ananth[2], Velankanni Cyril Raj[3]

[1]*Department of CSE, GMR Institute of Technology, Rajam, Andhra Pradesh, India*
[2]*Department of CSE, Sri Krishna College of Engineering and Technology, Coimbatore, India*
[3]*Department of CSE, Dr.M.G.R Educational and Research Institute, Chennai, India*
✉ *E-mail: rcristin2015@gmail.com*

Abstract: Forgery detection from the images is gaining remarkable interest as there are a lot of editing tools that enable to cause edition with manipulation or removal of the objects from the images. This study proposes a new forgery detection scheme that is based on the supervised learning approach. The supervised learning is brought about by using the support vector neural network and the optimisation is enabled using the fruit fly optimisation algorithm. Initially, the images are fed to the texture descriptor and the face is detected using the Viola–Jones algorithm. The face detected images are subjected to the feature extraction using the Gabor filter + wavelet + texture operator and the features are concatenated to present the input to the classifier. Then, the classifier which is trained using the fruit fly optimisation classifies the features to detect the presence of the manipulation. The performance of the proposed scheme is evaluated with the existing methods for the evaluation metrics accuracy, sensitivity, and specificity using two datasets, namely DSO-1 and DSI-1. The analysis shows that the proposed scheme attained an accuracy of 0.9523, the sensitivity of 0.94, and the specificity of 0.9583, which are greater when compared to the existing methods.

1 Introduction

The present-day technology has a huge impact on our day-to-day life due to the advances in the field of multimedia, internet, and imaging systems [1]. The users of the internet share a number of the multimedia contents, namely audios, videos, text, images in the social Medias daily, but accessing the contents with good resolution is a challenge [2]. The usage of the cheaper digital cameras and the user-friendly software affects the purity of the images. The security is a major concern that emerged long before [3]. The widespread availability of the applications has led to the availability of the editing software that led to the editing of the software in various ways [4]. The impact of the editing software led to the inability of detecting the manipulations present in the images because there are no enviable traces. The process of manipulating the images is termed as forgery and the existence of forgery led to the development of the digital forensics [5]. The digital forensics intends to detect the artefacts present in the images as a result of the forgery and to relieve the crime of manipulating through the recovery and investigation over the manipulated contents. Thus, the digital forensics leads to the authentication of the image [6]. Therefore, the image forgery detection and forensics identify if the image is edited or manipulated to ensure the trustworthiness [7, 8].

The field of forensics [9] classifies the image manipulation methods as any of the methods described below. The methods are cloning (copy–paste), splicing, erasing, and retouching. The splicing is the process of transferring a portion of the image into another image that can be done either through cut and paste operation or through the usage of other techniques such that there is a perfect match in the gradients of the target image [10, 11]. The operation copy–paste is same as splicing, but the portion of the image is transferred within the image. The process of retouching cannot be defined clearly but is something similar to that of blurring the portion of the image, recolouring, and applying filters [12, 13]. The image forgery is carried out in any of the following ways. It is possible to generate number of images through copying and pasting the portions of an image on another image [14]. Likewise, the portion of the image could be modified with some other portions of the same image, which is termed as healing, cloning, or retouching [14]. Moreover, the portions of the image are replaced with the image and the removed portion of the image is changed by region filling methods, namely image inpainting [8]. Thus, there are a diverse number of manipulating methods that offer a perfect image without any doubts of manipulation [15].

Image forgery detection is the process of detecting the presence or the absence of the manipulation in a given image [10]. There are two types of detecting forgeries in the image, one is the active forgery detection and another one is the passive forgery detection [8]. The former one is the non-blind approach that detects the information inserted in advance from the digital image. The examples of the active forensics are digital watermarks, signatures and so on, authenticating the images [16]. If the embedded image is found to be different from the original one then, the image is addressed as being tampered and image watermarking [17] is the famous method in the active forensics. In this image watermarking, the hidden message was inserted at the time of recording and it is unsheathed during the verification step [18]. In the passive forgery detection approach, there is no trace of the manipulation of the image and hence, it is a challenge [8]. Researchers are concentrating on the passive methods of detection [3]. These manipulations are determined using the alterations present in the statistical patterns obtained from image doctoring. The feature consistency of an image is determined using the watermark or signature by employing the blind methods [19]. Moreover, the image forgery detection techniques are grouped as block-based and keypoint-based techniques [20]. The former one divides the image into square blocks and the features of the square block are extracted following the image comparison so that to ensure the similarity [3], but the later method extracts the features of the image based on the high entropy followed by the image comparison of the image to identify the region duplication forgery detection [8].

To overcome the challenges in the existing forgery detection techniques, this paper proposes the forgery detection scheme using the illumination-based texture descriptor and fruit fly optimisation algorithm-support vector neural network (FOA-SVNN)-based classifier for classifying the images as forgery and non-forgery

images. Initially, the images are fed to the texture descriptor and the face is detected using the Viola–Jones algorithm which effectively determines the face in the image. The Viola–Jones algorithm is efficient and fast in detecting the face and the computational speed is reported in milliseconds. The face detected images are subjected to the feature extraction using the Gabor filter + wavelet + texture (GWTM) operator and the features are concatenated to present the input to the classifier. The Gabor filters and the wavelet transforms are advantageous in the GWTM operator as they preserve the facial features. Finally, the proposed FOA-SVNN classifier classifies the features to detect the presence of the manipulation which is trained using the fruit fly optimisation. The proposed scheme of classification is found to be effective in classifying the images such that if there are manipulations, the classifier reports are forged or otherwise, the image is reported as the original image. The importance of the proposed algorithm is that the computational speed is high and the best solution converges to the global optimal solution. The process of transformation to the code format is simple and easy while consuming less time. The training algorithm used is the optimisation algorithm, fruit fly optimisation algorithm (FOA), which improves the converging time and enhances the accuracy of the classification.

The contribution of the paper is *FOA-SVNN:* The proposed algorithm is the FOA-SVNN classifier that uses the FOA to train the support vector neural network (SVNN) such that the images are classified based on the availability of the forgery.

The paper is organised as follows: Section 1 presents the introduction of this paper, the existing works are deliberated in Section 2. The proposed method is detailed in Section 3, Results are discussed in Section 4 to prove the superiority of the proposed method. Finally, Section 5 concludes the paper.

2 Motivation

This section provides a deep introduction of the existing methods in detecting the forgeries and elaborates the challenges.

2.1 Related works

The related works are classified into five categories, namely (i) learning-based techniques, (ii) thresholding-based techniques, (iii) descriptors-based techniques, (iv) region-based techniques, and (v) lighting-based techniques.

2.1.1 Learning-based techniques: Carvalho *et al.* [1] proposed a forgery detection method used the inconsistencies in the illumination colour of the images. The forgery detection strategy discussed in [1] was the machine learning and it required very less user interaction. The shortcoming is that this technique is suitable for the images with two or three faces, but the advantage is that the method operates without the need of the experts' interaction. The proposed forgery detection method is suitable for detecting forgery images from number of faces. Schetinger *et al.* [13] offered a platform to inhale the forensic scenario in the fields of composition and digital image forensics. The benefit was that they were tuned properly for the specific case, but lacked the theoretical structure. Shen *et al.* [21] have proposed the passive splicing detection technique using textural features based on the grey level co-occurrence matrices, namely TF-GLCM. Here, the classification was performed by support vector machine (SVM). The TF-GLCM was better than state-of-the-art techniques with lower-dimensional feature vector.

2.1.2 Thresholding-based techniques: Jeronymo *et al.* [11] proposed an error level analysis for the detection of forgery in the digital images that seems to be lossy when compressed. The noise is removed automatically using the soft-thresholding. The method is effective in removing the noise and other high-frequency regions and stands effective without any blurring effect, but the weakness is regarding the compression that injected heavy noise in the compressed image. The proposed method reduces the noise in the compressed image since, it takes the advantages of both FOA and SVNN. Hayat and Qazi [2] introduced a forgery detection method based on the discrete wavelet transform (DWT) and discrete cosine transform (DCT) that are employed for reducing the features. The DCT is applied to the individual blocks obtained as a result of DWT and the comparison is made depending on the correlation coefficient. The merit of the method is regarding the viability of the copy/move and splicing-based forgeries, but offered bad effects in the presence of the occlusion. The proposed method effectively finds the forgery even if the occlusion is presented in the image. Hu *et al.* [18] proposed an image forgery detection method that was effective in determining the tampered foreground or background image through image watermarking and alpha mattes. The advantages of the method were that the foreground images were detected accurately and detected the forgeries, and the thresholds employed enabled the method to be suitable for the practical applications. However, the method is applicable for detecting the forgeries only in the copy–paste images. The proposed method is applicable for any type of images.

2.1.3 Descriptors-based techniques: Farooq *et al.* [5] proposed a generic passive image forgery scheme depending on the spatial rich model along with the local binary pattern (LBP). The method is found to be more accurate and less complex such that only less number of sub-models is employed. Even though the complexity is low, the dimension of the features can be reduced only through developing the better models. In the proposed method, the process of transformation to the code format is simple and easy while consuming less time. Liu *et al.* [22] have introduced an integrated algorithm by utilising the Joint Photographic Experts Group (JPEG) features and local noise discrepancies for detecting the fraud practices, such as copy–move (CM) and splicing forgery in a digital image. This technique was effective on detecting both CM and splicing forgery.

2.1.4 Region-based techniques: Bhartiya and Jalal [8] proposed a method to detect the forgeries in the JPEG images using the feature-based clustering. The method is found to be accurate and fast, but the failure is that the proposed detection algorithm exhibits huge false positives. Moreover, there is no proper balance between the accuracy and the execution cost of the method. FOA utilised in the proposed method improves the converging time and enhances the accuracy of the classification Mahmood *et al.* [3] proposed an effective technique for detecting the forgery based on the region duplication in the digital images. The features extracted from the overlapping images were employed to detect the region duplication. The method minimises the false detection and reduces the dimension of the features such that the execution time of the algorithm is minimised, but it is revealed that the block size of the overlapped blocks is dependent on the duplicate region size.

2.1.5 Lighting-based techniques: Peng *et al.* [23] have presented an optimised 3D lighting estimation technique by integrating a general surface reflection model. This reflection model was more accurate. Therefore, it attained the maximum lighting estimation accuracy and reliable discrimination performance.

2.2 Challenges

- The most significant function in CM forgery is that either it hides a region within an image or injects a new pattern in the image. CM alteration is carried out based on any one of the following modifications, such as transformations in geometry including the rotation, scaling, improving the intensity like brightness or darkness, linear filtering that includes blurring and noise effect, or compression like JPEG encoding. However, the process of detecting the duplicated regions in the image is a hectic challenge [9].
- In general, the digital images are presented in the JPEG format and the image can be recompressed using the JPEG after the completion of the manipulation. Prior to the detection of the manipulation in the image, it is essential to determine if the image is a JPEG recompressed image or not, which is a hint.

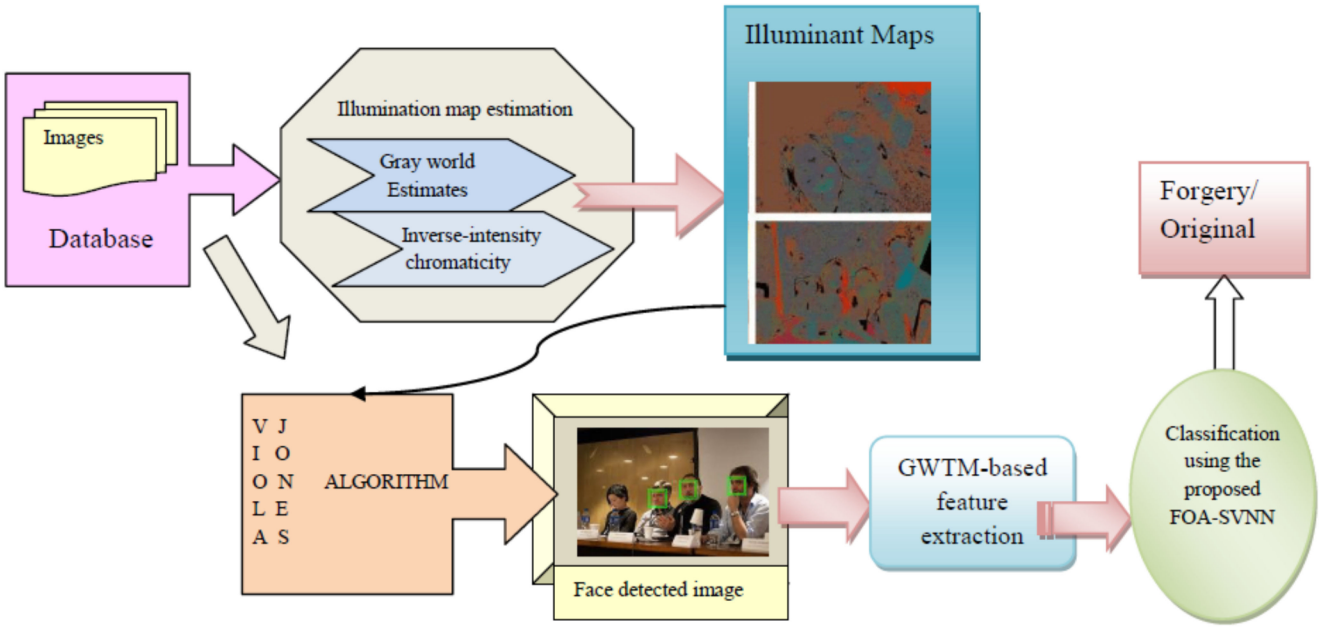

Fig. 1 *Block diagram of the proposed scheme of forgery detection using the FOA-SVNN classifier*

The major challenge is regarding the identification of the double quantisation in blocks of the image in order to detect the manipulated area in the JPEG image. Thus, it is concluded that the detection of the quantisation effects in the image for different JPEG compression qualities including the first and the second compression is the standing challenge of the time [8].

- The replication of the object at the conditions, i.e. the contents created from it cannot authenticate the image. The duplicated regions obtained are highly blended at the required locations such that the identification made visually is impossible [3].

3 Proposed scheme of detecting forgery using the illumination-based texture descriptor and the proposed FOA-SVNN

The need for the automatic forgery detection scheme arises in the present scenario, where the trustworthiness of the images is not assured. The availability of a large number of the user-friendly software enables the editing and the manipulation of the images that affect the originality of the images leading to the illegal forgery. This paper presents a forgery detection scheme using the proposed FOA-SVNN classification. The features are presented for classification and the features are extracted using the GWTM. Initially, the image is subjected to the illumination map using two estimates. In the second step, the Viola–Jones algorithm is employed for detecting the faces in the image and then, finally, the face detected images are subjected to the feature extraction using the GWTM. The GWTM presents three features that are concatenated to form the three inputs to the SVNN classifier, which classifies the features and provide the information if it is forgery or non-forgery image. Fig. 1 shows the block diagram of the proposed method. Let us consider an image I present in the database such that the image is subjected to detect the forgery using the proposed classifier.

3.1 Illumination map estimation

The estimation of the illumination map [1] is depicted in this section. The segments of super pixels are formed from the input image using the Felzenszwalb and Huttenlocher [24]. The colour of the illumination is determined for all the individual superpixels. In order to estimate the illumination, the colour estimators employed are inverse-intensity chromaticity space and the grey world estimates.

3.1.1 Grey world estimates: As per the assumption made in the existing work [25], grey is the average scene colour and hence, it is clear that any deviation from the average grey is the cause of illumination. Let us denote the red, green, blue (RGB) colours of a pixel centred at z to be

$$F(z) = [F_{\mathrm{R}}(z), F_{\mathrm{G}}(z), F_{\mathrm{B}}(z)] \tag{1}$$

As an assumption, it is assumed to be a truly diffuse reflection and linear camera response that is given below:

$$F(z) = \int_{\eta} e(\lambda, z) r(\lambda, z) L(\lambda)\, \mathrm{d}\lambda \tag{2}$$

where η represents the visible light spectrum, λ is the wavelength of light, $e(\lambda, z)$ refers to the illuminant spectrum, $F_{\mathrm{G}}(z)$ represents the surface reflectance. The sensitivity of the camera to colour is denoted as $L(\lambda)$. The world grey hypothesis is expanded using the manipulation of three parameters, namely the order of the derivative m, norm of Minkowski ρ, and smoothing parameter, σ. The colour of the illuminant e is denoted as

$$K\, e^{m,\rho,\sigma} = \left(\int \left| \frac{\partial^m F^{\sigma}(z)}{\partial z^m} \right|^{\rho} \mathrm{d}z \right)^{\frac{1}{\rho}} \tag{3}$$

where z denotes the pixel coordinate, K defines the scaling factor, and $|\cdot|$ determines the absolute value. The differentiation is denoted as ∂ and $F^{\sigma}(z)$ represents the observed intensities at a position z that is smoothened using the kernel σ. The parameter e is computed individually for the colour channel and the derivative operator improves the robustness.

3.1.2 Inverse-intensity chromaticity: In contrary to the grey world estimates, the assumption made in this approach exhibits the diffuse and specular reflectance. Consider (1) that shows the RGB pixel of the image. The function is designed as

$$F(z) = [F_{\mathrm{R}}(z), F_{\mathrm{G}}(z), F_{\mathrm{B}}(z)]^{\mathrm{T}} \tag{4}$$

The relationship between the function $F(z)$, chromaticity of the colour channel $\gamma_L(z)$, chromaticity of the illuminant channel χ_L is given by

$$\gamma_L(z) = n(z) \times \frac{1}{\sum_{j \in \{\mathrm{R,G,B}\}} F_j(z)} + \chi_L \tag{5}$$

The function $n(z)$ determines the geometric influences, which cannot be calculated analytically, but undergoes an approximate solution. Thus, the only way of determining the illuminant colour is through the y-intercept χ_L that is determined by analysing the pixels.

3.2 Face extraction using the Viola–Jones

The importance of using the Viola–Jones algorithm [26] for detecting the face is that the algorithm is efficient and fast in detecting the face and the computational speed is reported in milliseconds. Initially, the black pixels are marked and subtracted from the white pixels and the results are compared with the threshold value such that the features are identified based on the criterion. The steps in Viola–Jones are given as

(a) *Haar-like features:* The Haar-like features determine the black and the white portions of the image that uses a rectangle around the face.
(b) *Formation of the integral image:* The integral image is formed by summing the pixel value of the individual pixels with the pixel values of its neighbours. In other words, the value of the individual pixel is formed by summing the pixel value of the neighbouring four pixels that is accumulated in the rectangle.
(c) *Adaboost machine-learning method:* The Adaboost is the machine-learning method employed for detecting the face and it follows the bagging concept. The importance of the Adaboost algorithm is to choose small features of the face such that the computation becomes easy and fast. The AdaBoost algorithm provides the highly significant features by neglecting the unnecessary background.
(d) *Cascade classifier for concatenating the features:* The cascade classifier comprises of a number of the classifiers that allows the selection of the face image. Each of the sub-windows is send to the classifiers such that to determine whether the sub-window possess the face or not.

Thus, the Viola–Jones algorithm to detect the face from the image is performed successfully using the above-detailed steps. The original image is employed for detecting the face using the Viola–Jones.

3.3 Feature extraction using GWTM

The input face-detected images are subjected to the feature extraction using the GWTM operator [20]. The images are fed to the wavelet transform and the Gabor filter, among which the wavelet images are generated using the wavelet transform and features based on the orientation and the frequency are determined using the Gabor filters. The output from the wavelet and the Gabor filters is fed to the LBP model with the input image. The analysis of the texture is analysed for the input image that transforms the input image into an array. The output from the LBP is subjected to the histogram analysis such that the histogram yields the global appearance of the image. Fig. 2 shows the feature extraction steps using GWTM.

The Gabor filters and the wavelet transforms are advantageous in the GWTM operator used for the feature extraction as they preserve the facial features. The GWTM considers the approximation coefficient for extracting the features as they possess the tendency to reveal the facial features in a different scale. Moreover, the Gabor filters yield the frequency information and they preserve spatial and frequency information of the image. The Gabor filters possess the phase and the magnitude information and the magnitude of the image gains significance as they preserve the edge information effectively.

3.3.1 Wavelet transforms: The face detected images are subjected to the wavelet transforms that enable to analyse the stationary and the non-stationary images. The wavelet transform decomposes the image into a set of functions, termed as the wavelets. The wavelet transform enhances the precision of the

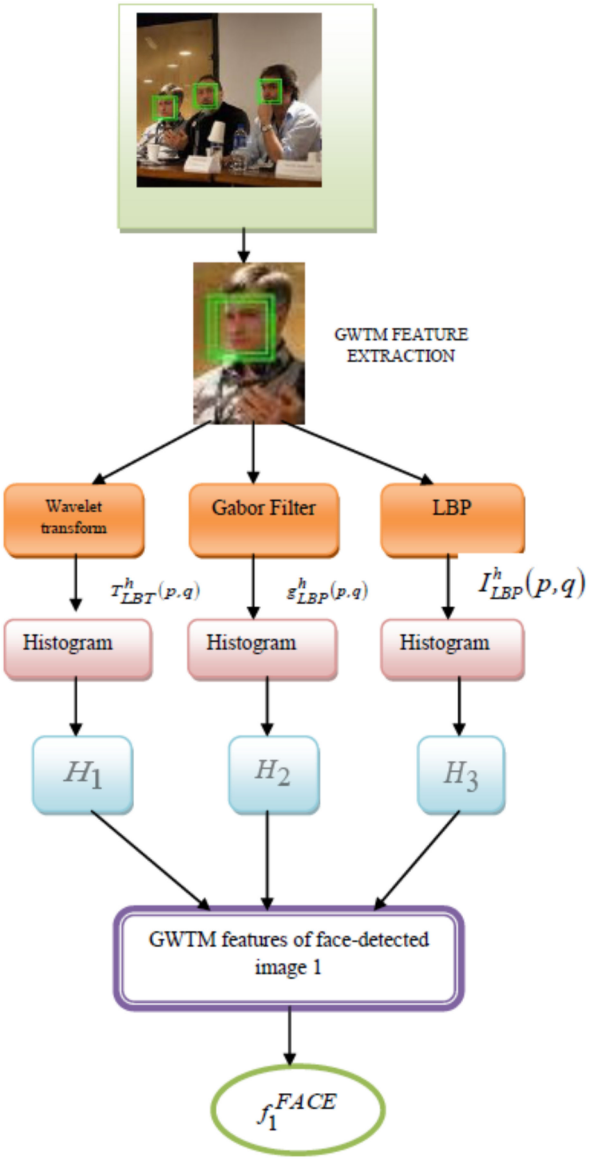

Fig. 2 *GWTM feature extraction*

forgery detection process and the 2D wavelet transform is given below:

$$T^h(p,q) = \frac{1}{\sqrt{PQ}} \sum_{p=1}^{P-1} \sum_{q=1}^{Q-1} I^h(p,q) * \varphi_{k,l}(p,q) \tag{6}$$

where $I^h(p,q)$ denotes the high-resolution input image and $\varphi_{k,l}(p,q)$ specifies the scaling function.

3.3.2 Gabor filters: The Gabor filtering is the second step of the GWTM [20] and the purpose of using Gabor filtering is that the Gabor filters conclude the spatial locality, orientation, and detects the edge of the face detected image. Moreover, Gabor filters provide the texture features and it is given below:

$$X^h_{k,l}(s) = \frac{\| D_{k,l} \|^2}{\varepsilon^2} \times \mathrm{e}\left(-\frac{\left(\| D_{k,l} \|^2 - \| s \|^2\right)}{2\varepsilon^2}\right)\left[\mathrm{e}^{p.D_{k,l}s} - e^{-\frac{\varepsilon^2}{2}}\right] \tag{7}$$

where k and l are the orientation and scale of the Gabor filters, respectively, and s represents the spatial position. $D_{k,l}$ is used to determine the magnitude value as

$$D_{k,l} = \frac{D_{\max}}{\psi} \tag{8}$$

where ψ refers to the frequency ratio. The Gabor filter output is the convolution of the face detected the image $I^h(p,q)$ and $X^h_{p,q}(s)$, as given below:

$$g^h_{k,l}(p,q) = I^h(p,q) * X^h_{p,q}(s) \tag{9}$$

The complex form of the Gabor filter output is represented in (10) that is employed for recognising the image forgery.

$$g^h_{k,l}(p,q) = M_{k,l}(p,q) \times e^{i\varphi(p,q)} \tag{10}$$

Depending on the available frequency channels and the directions, the Gabor filter extracts the local features from the face detected image. The image features from the Gabor filters are obtained through magnitude, phase features of the Gabor, real and imaginary features of the Gabor. The further processes in this section are preceded using the real part of the Gabor magnitude.

3.3.3 Extraction of the local features using LBP: The output from the wavelet and the Gabor features are applied to the LBP for extracting the local features. The input to the LBP is the grey level image such that the LBP pattern does not change based on the changes in the monotonic greyscale image. From the input grey image, the centre pixel is fixed as, t_c and then, the comparison is made between the centre pixel and the neighbouring pixels

$$\mathrm{LBP}(u_c, v_c) = \sum_{i=0}^{7} V(t_i - t_c) \times 2^i \tag{11}$$

$$V(b) = \begin{cases} 1, & b \geq 0 \\ 0, & b < 0 \end{cases} \tag{12}$$

where t_i indicates the neighbouring pixel to the centre pixel t_c. During the comparison, whenever the pixel value of the neighbouring pixel is greater than the centre pixel, the neighbouring pixel gains the value '1' or otherwise; the pixel value becomes '0' as explained in (12). Thus, a new image with '1's and '0's is formed and is used as the threshold image and the LBP pattern is formed by transforming the binary threshold image into the corresponding decimal value. The texture classification is carried out by using the input face detected an image, approximation coefficients of the wavelet transform, and the real part of Gabor magnitude. The LBP operator for the approximation coefficients is given as

$$T^h_{\mathrm{LBT}}(p,q) = \mathrm{LBP}\left[T^h(p,q)\right] \tag{13}$$

where $T^h(p,q)$ denotes the output of the wavelet transform and $T^h_{\mathrm{LBT}}(p,q)$ indicates the LBP of the wavelet transform. Applying the face detected image to the LBP gives

$$I^h_{\mathrm{LBP}}(p,q) = \mathrm{LBP}\left[I^h(p,q)\right] \tag{14}$$

where $I^h_{\mathrm{LBP}}(p,q)$ indicates the LBP of the HR image. The LBP of the Gabor filter output is

$$g^h_{\mathrm{LBP}}(p,q) = \mathrm{LBP}\left[g^h_{k,l}(p,q)\right] \tag{15}$$

where $g^h_{k,l}(p,q)$ defines the Gabor filter output from the HR images and the LBP of the Gabor filter output is given as $g^h_{\mathrm{LBP}}(p,q)$.

3.3.4 Histogram representation: The histogram displays the pixel values of the input image and the image representation portrays the frequency of the grey levels in the image. The input to the histogram representation is the results of the wavelet transform, Gabor filter, and the face detected image. The results of the histogram are represented as

$$f_1^{\mathrm{FACE}} = \left\{H_1^1 \| H_1^2 \| H_1^3\right\} \tag{16}$$

where H_1^1 represents the histogram output corresponding to the wavelet transform of the face detected image 1, H_1^2 denotes the output of histogram for the Gabor filter result of the face detected image 1, and H_1^3 implies the histogram output for the input face detected image 1. f_1^{FACE} is the GWTM feature of the face-detected image 1. When there exist 256 bins in the LBP histogram then, the feature-length of GWTM is 768. If there are three faces in the image then, the GWTM features are given as

$$f_2^{\mathrm{FACE}} = \left\{H_2^1 \| H_2^2 \| H_2^3\right\} \tag{17}$$

$$f_3^{\mathrm{FACE}} = \left\{H_3^1 \| H_3^2 \| H_3^3\right\} \tag{18}$$

where f_2^{FACE} and f_3^{FACE} corresponds to the GWTM features of the face detected images 2 and 3, respectively. H_2^1, H_2^2, and H_2^3 are the histogram outputs corresponding to the wavelet transform, Gabor filter, and the face detected input image 2. H_3^1, H_3^2, and H_3^3 are the histogram outputs of the face detected image 3 that corresponds to the wavelet transform, Gabor filter, and the face detected input image 3.

3.4 Feature input for classification using the proposed classifier

The feature vector obtained as a result of feature extraction using the GWTM is $f_t^{\mathrm{FACE}} = \left\{f_1^{\mathrm{FACE}}, \ldots, f_u^{\mathrm{FACE}}, \ldots, f_t^{\mathrm{FACE}}\right\}$. Each of the features is of dimension, (1×768) and the features are presented to the proposed classifier for classifying the image. The first feature of the proposed classifier is formed by concatenating the features of GWTM, f_1^{FACE} and f_2^{FACE}, to yield

$$f_1^* = \left[f_1^{\mathrm{FACE}} \| f_2^{\mathrm{FACE}}\right] \tag{19}$$

The second input to the proposed classifier is formed by merging the features f_2^{FACE} and f_3^{FACE} as given below:

$$f_2^* = \left[f_2^{\mathrm{FACE}} \| f_3^{\mathrm{FACE}}\right] \tag{20}$$

The third feature vector given as the classifier input, which is obtained by concatenating the features f_3^{FACE} and f_1^{FACE}, is presented as below:

$$f_3^* = \left[f_3^{\mathrm{FACE}} \| f_1^{\mathrm{FACE}}\right] \tag{21}$$

In general, the feature concatenation is given as

$$f_a^* = \left[f_u^{\mathrm{FACE}} \| f_v^{\mathrm{FACE}}\right], \quad \forall u \text{ and } v;\ 0 \leq a \leq b \tag{22}$$

where b refers to the total number of the possible training vectors generated using the GWTM features of the face detected images. f_u^{FACE} and f_v^{FACE} are the GWTM features extracted from the uth and the vth face detected images. Thus, the training feature vector used as input to the proposed classifier is

$$f^* = \{f_1^*, f_2^*, f_3^*, \ldots, f_a^*\} \tag{23}$$

3.5 Classification of the image using the proposed FOA-SVNN classifier

The proposed FOA-SVNN is the SVNN that is trained using the FOA. The importance of the proposed algorithm is that the computational speed is high and the best solution converges to the global optimal solution. The process of transformation to the code format is simple and easy while consuming less time. The FOA [27] is the optimisation mechanism that exhibits the fruit search behaviour of the flies based on the smell and perception. The

osphresis and vision of the fruit flies are essential for the smell and they are capable of sensing the smell even when they are 40 km apart. Once the location of the food is near, they use their vision to find the food. The execution speed is faster when a large number of the fruit flies search for food through a stable route. The FOA offers simple computational process, and it offers an easy transformation of the concept to the program code and it is easy to understand. The SVNN is employed for classifying the image features. It converges to the global minima rather than the local minima, and SVNN never considers the noise, unlike the artificial neural networks. Due to these reasons, the SVNN is employed for the classification.

3.5.1 Architecture of SVNN: The SVNN [28] comprises of the input layer, hidden layer, and the output layer, as shown in Fig. 3. SVNN is generated by introducing the eigenvalue decay in the neural networks (NNs) based on the same principles of SVM hence, the classification margin is improved. The use of SVM with non-linear kernels requires a prohibitive computational cost; since its decision function needs a summation of non-linear functions which demands a large amount of time when the number of support vectors is big. Therefore, a maximal margin NN can be a suitable option, since it can offer a fast non-linear classification with good generalisation capacity. The input to the SVNN is the features extracted from the images that are obtained as a result of concatenating the GWTM features. The features fed to the proposed FOA-SVNN classifier trains the network such that the forgery detection is made. Initially, 'a' features are fed into the input layers. These features are multiplied with W_2 and are fed into the hidden layer. After that, bias W_3 is added with these features and the weight of the output layer is added. Finally, at the output layer, the bias of the output layer is added and the classified results are obtained. Fig. 3 presents the architecture of the classifier for classifying the *i*th image present in the database. Thus, the output of SVNN is given below:

$$O^{\text{SVNN}} = W_1 \cdot \log\text{sig}\left[\left(\sum_{l=1}^{a} f_l^* * W_2\right) + W_3\right] + W_4 \tag{24}$$

where W_1 is the weight between the hidden layer and the output layer, W_2 is the weight between the input layer and the hidden layer, W_3 is the bias of the hidden layer, and W_4 is the bias of the output layer. l indicates the number of features corresponding to a face detected image for identification as forged or not. If the classifier output is '1', then, the image is found to be the forged images or else, the image is the true image. If anyone of the features in $f^* = \{f_1^*, f_2^*, f_3^*, \ldots, f_a^*\}$ deviates from the original image then, the SVNN classifier recognises the image as the forged.

3.5.2 Training using the FOA: The steps involved in the FOA are depicted below:

Step 1: Initialisation: The population of the fruitfly is initialised in the first step and the initial position of the fruit flies is said to be in the initial position. Let the initial position of the fruitfly group be $(x_{\text{ini}}, y_{\text{ini}})$.

Step 2: Random search: The random search for the direction and distance is carried out using the osphresis of the individual fruitfly. The direction and distance of the fruitfly are updated based on the following equations:

$$x_i = x + Rand \tag{25}$$

$$y_i = y + Rand \tag{26}$$

where x_i and y_i are the position and the direction of the *i*th fruitfly, respectively, x and y are the initial positions of the fruitfly, and *Rand* is the random value.

Step 3: Calculation of location of food: The location of the food is not known and hence, the distance to the origin (d_i) is determined initially. Followed by the calculation of the distance,

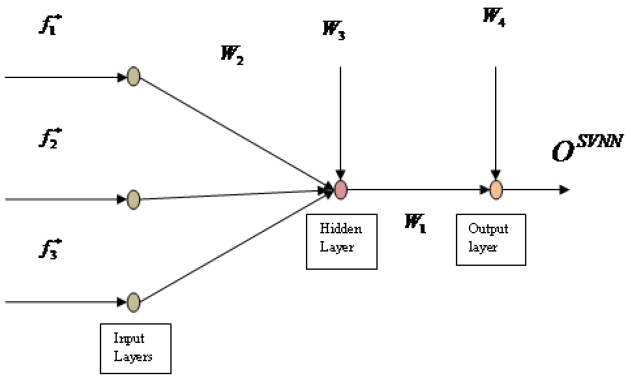

Fig. 3 *Architecture of SVNN*

the concentration of smell judgement value is evaluated by taking the reciprocal of the distance

$$d_i = \sqrt{x_i^2 + y_i^2} \tag{27}$$

The smell concentration is computed as

$$J_i = \frac{1}{d_i} \tag{28}$$

Step 4: Compute the objective function: The objective function is based on the smell concentration judgment value and the maximum smell is considered as the fitness. The objective function is given by

$$S_i = \delta_{\max} + \delta_{\min} + \frac{C}{n}\sum_{k=1}^{n} |O_k - O_k^*| \tag{29}$$

where S_i corresponds to the objective function or the smell concentration function of the *i*th fruitfly that depends on the judgment value of smell concentration J_i. The value of the objective function aims at the maximum value of the fitness, where O_k^* refers to the class value of the ground data, O_k denotes the estimated output of the SVNN, and n indicates the total number of the training samples. The regularisation factor is denoted as C.

$$\delta = \text{eigen}\left(W \times W^{\text{T}}\right),$$
$$\delta_{\max} = \max(\delta), \quad \delta_{\min} = \min(\delta)$$

Step 5: Compute the best fruitfly: The fruitfly with the maximum smell concentration is considered as the best fruitfly, as defined below:

$$B_f^{\text{best}} = \text{Max}(S_i) \tag{30}$$

The maximum value of the smell is retained such that the fruitfly flies towards the location that corresponds to the maximum smell using the vision

$$\text{Max}(S) = \text{best smell} \tag{31}$$

The position and the direction of the fruitfly that provides the maximum value of the smell is given by

$$x = x_{i-\text{best}} \tag{32}$$

$$y = y_{i-\text{best}} \tag{33}$$

Step 6: Termination: The steps from 2 to 5 are repeated and if the smell is greater when compared with the smell judgment value of the previous iteration, then keep the smell judgment value of the previous iteration or otherwise continue with the search process Fig. 4 shows the pseudo code of FOA.

	FOA
1	Input : Initial position (x_{ini}, y_{ini})
2	Output: Best positions x_{i-best} and y_{i-best}
3	
4	Begin
5	Initialization:
6	Read the initial position and direction of the fruitfly.
7	Enable the random search.
8	Compute the location of the food.
9	Evaluate the objective function.
10	Determine x_{i-best} and y_{i-best}.
11	Iterate steps 7-10.
12	End

Fig. 4 *Pseudo-code of FOA*

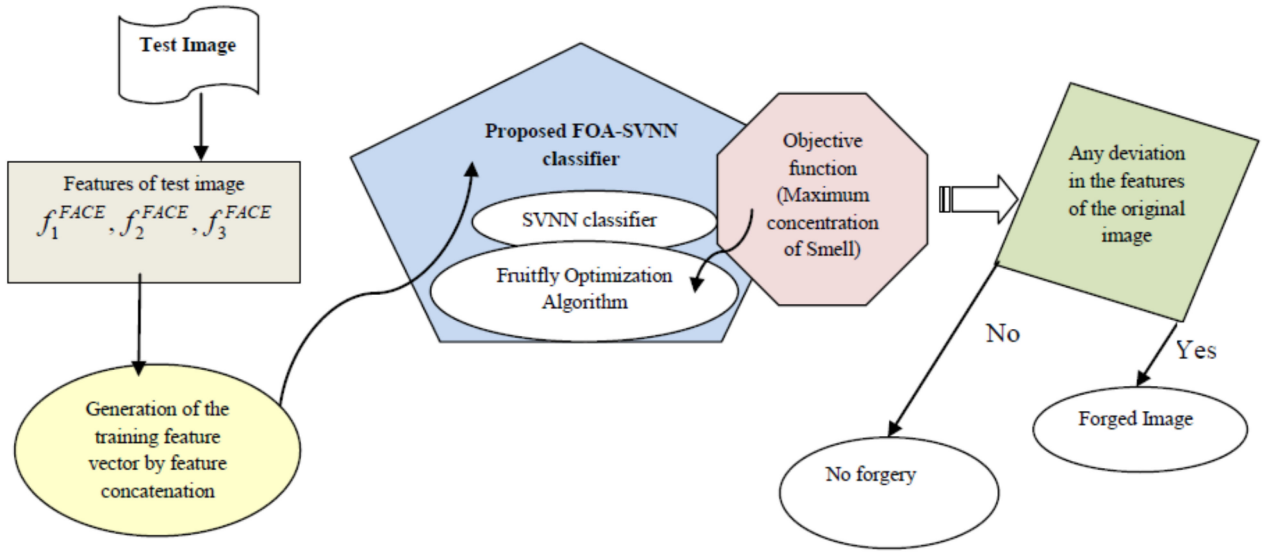

Fig. 5 *Detecting the forgery using the proposed classifier*

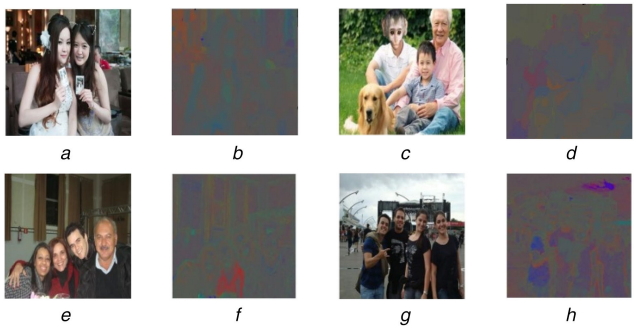

Fig. 6 *Sample results of the experiment using the proposed algorithm*
(a) Normal image 1, ***(b)*** Greyscale of normal image 1, ***(c)*** Fake image 1, ***(d)*** Greyscale of fake image 1, ***(e)*** Normal image 2, ***(f)*** Greyscale of normal image 2, ***(g)*** Fake image 2, ***(h)*** Greyscale of fake image 2

3.5.3 Testing phase: Whenever the new image arrives at the classifier for detecting the forged image, the classifier classifies and provides the output. If the output is one then, the image is said to be forged or else, the image is the original image without any manipulation. Fig. 5 shows the testing phase of the proposed classifier.

4 Results and discussion

This section deliberates the results and discussion of the proposed method in order to prove the effectiveness of the proposed method.

4.1 Experimental setup

The experimentation of the proposed technique of detecting the forgery is done in the system with 2 GB RAM, Intel core processor, Windows 10 Operating System. The technique is implemented using the software tool MATLAB.

4.2 Dataset description

For the experimentation, two datasets [1], such as DSO-1 and DSI-1, are employed.

(i) *DSO-1:* DSO-1 (dataset 1) consists of 200 indoor and outdoor images with an image resolution of 2048 × 1536 pixels. The dataset comprises of 100 original images and 100 forged images. The images are forged by including one or more individuals in the source image with one or more persons.
(ii) *DSI-1:* DSI-1 (dataset 2) consists of 50 images with 25 original and 25 doctored images taken from different websites on the Internet with different resolutions.

4.3 Experimental results

This section presents the sample results of the proposed method using two datasets that uses the image with and without forgery as shown in Fig. 6. The original images are shown in Figs. 6*a* and *e* and the fake images are shown in Figs. 6*c* and *g*, respectively. The greyscale images of the original images are given in Figs. 6*b* and *f* and the greyscale images of the fake images are given in Figs. 6*d* and *h*, respectively.

Step-by-step tracing of the proposed technique for normal image and a fake image was taken from both DSO-1 and DSI-1 datasets is given in Figs. 7–10.

4.4 Performance metrics

The metrics used for the experimentation are described in this section.

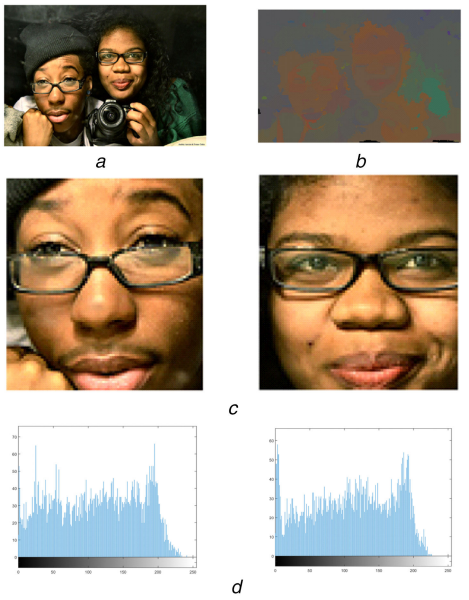

Fig. 7 *Step-by-step tracing of the proposed technique for a normal image taken from the DSO-1 dataset*
(a) Normal image, ***(b)*** Illumination map, ***(c)*** Face detected image, ***(d)*** GWTM-based feature extraction

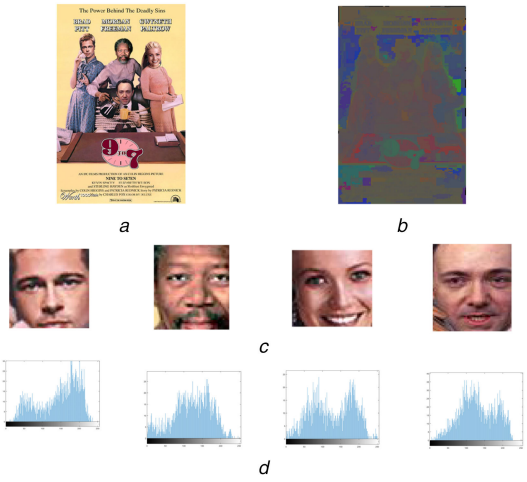

Fig. 8 *Step-by-step tracing of the proposed technique for a fake image taken from the DSO-1 dataset*
(a) Fake image, ***(b)*** Illumination map, ***(c)*** Face detected image, ***(d)*** GWTM-based feature extraction

4.4.1 Accuracy: It is the measure of correctness of the detection as given as

$$\mathrm{TPR} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{34}$$

where TP is true positive, TN is true negative, FN is false negative and FP is false positive.

4.4.2 Sensitivity: The sensitivity or the TP rate (TPR) is defined as the number of positives identified correctly

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{35}$$

4.4.3 Specificity: The specificity or the TN rate (TNR) is defined as the number of negatives identified correctly

$$\mathrm{TNR} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{36}$$

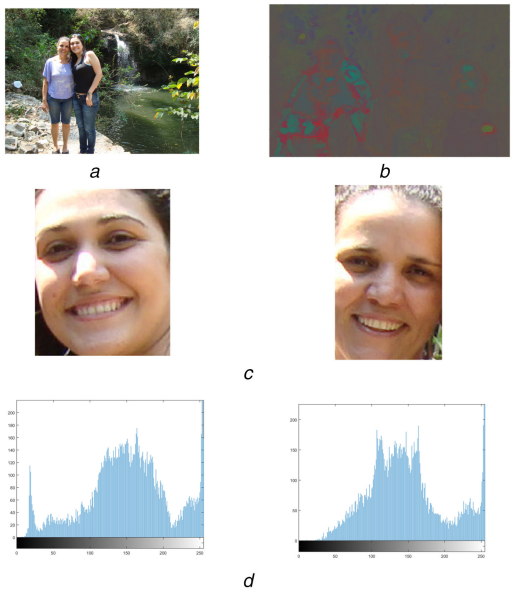

Fig. 9 *Step-by-step tracing of the proposed technique for a normal image taken from the DSI-1 dataset*
(a) Normal image, ***(b)*** Illumination map, ***(c)*** Face detected image, ***(d)*** GWTM-based feature extraction

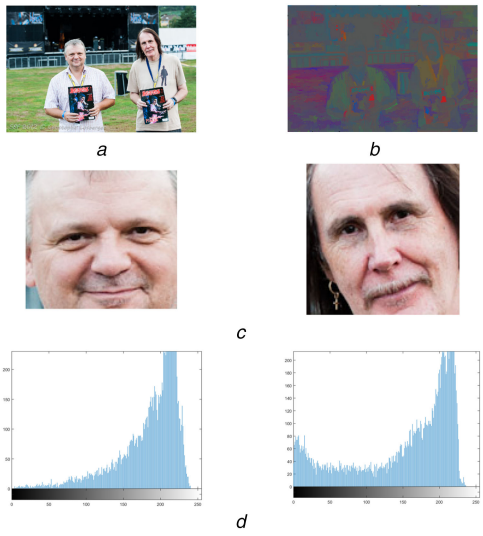

Fig. 10 *Step-by-step tracing of the proposed technique for a fake image taken from the DSI-1 dataset*
(a) Fake image, ***(b)*** Illumination map, ***(c)*** Face detected image, ***(d)*** GWTM-based feature extraction

4.5 Competing methods

The methods taken for comparison include the K-nearest neighbours (KNN) [29], NN [19], SVM [13], and SVNN [28] so that comparison is made with them to prove the superiority of the proposed work.

4.6 Comparative analysis

The comparative analysis of the methods is depicted in this section using two datasets.

4.6.1 Based on training percentage using dataset 1: Fig. 11*a* depicts the accuracy of the methods using the dataset1 concerning the training percentage. When *k*-value is 0.5, the accuracy of the proposed FOA-SVNN is 0.8421 which is 35.34% greater than the existing SVNN. When training percentage is 0.9, the rate of accuracy of the proposed method is 3.9% superior to SVM. It is clear from the discussion that the proposed FOA-SVNN offers a greater rate of the accuracy. Fig. 11*b* depicts the TPR of the

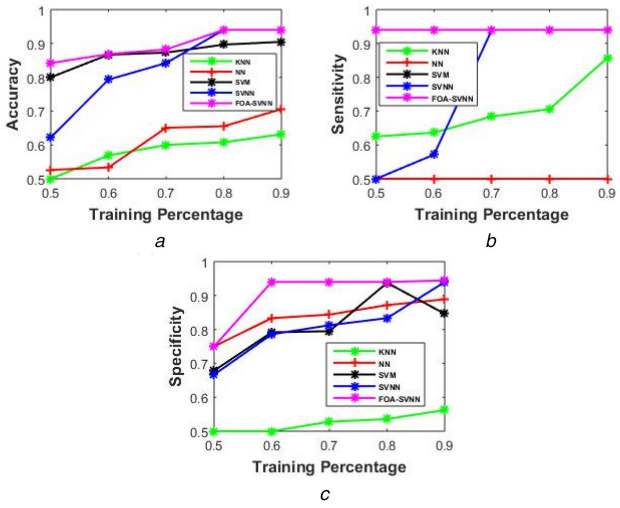

Fig. 11 *Analysis using database 1 based on training percentage*
(a) Accuracy ***(b)*** Sensitivity ***(c)*** Specificity

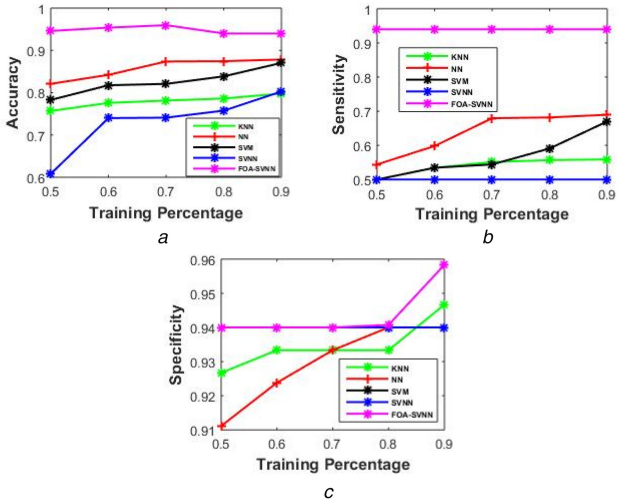

Fig. 12 *Analysis using database 2 based on training percentage*
(a) Accuracy ***(b)*** Sensitivity ***(c)*** Specificity

methods using the dataset1 concerning the training percentage. When k-value is 0.5, the TPR of the proposed method is 1.06% superior to existing SVM. When training percentage is 0.9, the rate of the TPR of the proposed method is 9.67% superior to KNN. Fig. 11c depicts the TNR of the methods using the dataset 1 concerning the training percentage. When k-value is 0.5, the TNR of the FOA-SVNN is 12.61% superior to SVNN. When training percentage is 0.9, the rate of the TNR of the FOA-SVNN 11.9% superior to SVM.

4.6.2 Based on training percentage using dataset 2: Fig. 12a depicts the accuracy of the competing methods using dataset 2 concerning the training percentage. When k-value is 0.5, the accuracy of the FOA-SVNN is 0.9459 which is greater than the other existing methods. When the training percentage is 0.9, the rate of the accuracy of the FOA-SVNN is 7.02% superior to NN. It is clear from the discussion that the proposed FOA-SVNN offers a greater rate of the accuracy. Fig. 12b depicts the TPR of the methods using dataset 2 concerning the training percentage. The TPR of the proposed FOA-SVNN is 0.94 when k-value is 0.5, which is superior to the TPR of other existing methods. When the training percentage is 0.9, the rate of the TPR of FOA-SVNN is 36.31% superior to NN. It is clear from the discussion that the proposed FOA-SVNN offers a greater rate of the TPR. Fig. 12c depicts the TNR of the methods using dataset 2 concerning the training percentage. When k-value is 0.5, the TNR of FOA-SVNN is 0.94. When training percentage is 0.9, the rate of the TNR is

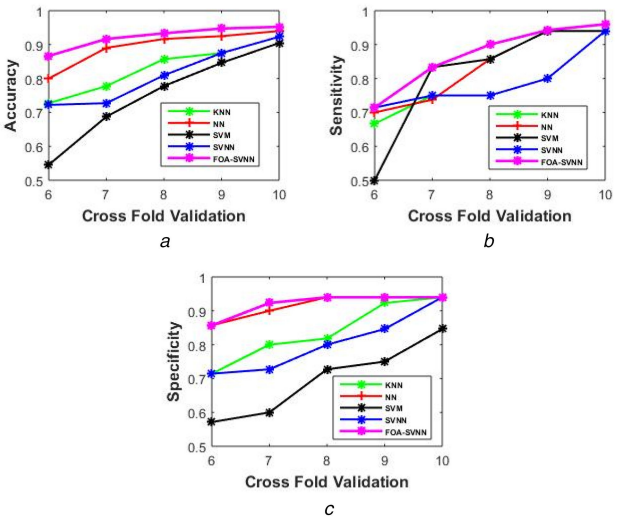

Fig. 13 *Analysis using database 1 based on cross-fold validation*
(a) Accuracy ***(b)*** Sensitivity ***(c)*** Specificity

found as 0.95833 for FOA-SVNN. It is clear from the discussion that the proposed FOA-SVNN offers a lower rate of the TNR.

4.6.3 Based on cross-fold validation using dataset 1: Fig. 13a depicts the accuracy of the methods using dataset 1 concerning the cross-fold validation. When k-value is 6, the accuracy of the FOA-SVNN is 19.16% superior to KNN. When k-value is 10, the rate of the accuracy is found as 0.9523 for the proposed FOA-SVNN. It is clear from the discussion that the proposed FOA-SVNN offers a greater rate of the accuracy. Fig. 13b depicts the TPR of the methods using dataset 1 concerning the cross-fold validation. The TPR of FOA-SVNN is 7.23% superior to KNN when k-value is 6. When k-value is 10, the rate of the TPR is found as 0.96 for FOA-SVNN. Fig. 13c depicts the TNR of the methods using dataset 1 concerning the cross-fold validation. The TNR of the proposed FOA-SVNN is 20% superior to KNN when k-value is 6. When k-value is 10, the rate of the TNR is found as 0.94 for FOA-SVNN. Thus, it is clear from the discussion that the proposed FOA-SVNN offers a lower rate of the TNR.

4.6.4 Based on cross-fold validation using dataset 2: Fig. 14a depicts the accuracy of the methods using dataset 2 concerning the cross-fold validation. The accuracy of FOA-SVNN is 0.48% superior to NN when k-value is 6. When the k-value is 10, the rate of the accuracy is found as 0.94 for KNN, NN, SVM, SVNN, and FOA-SVNN. It is clear from the discussion that the proposed FOA-SVNN offers a greater rate of the accuracy. Fig. 14b depicts the TPR of the methods using dataset 2 concerning the cross-fold validation. The TPR of FOA-SVNN is 0.9454 when k-value is 6, which is greater than the TPR of the existing methods. When k-value is 10, the rate of the TPR is found as 0.94 for KNN, NN, SVM, SVNN, and FOA-SVNN. It is clear from the discussion that the proposed FOA-SVNN offers a greater rate of the TPR. Fig. 14c depicts the TNR of the competing methods using dataset 2 concerning the cross-fold validation. The TNR of FOA-SVNN is 20% superior to KNN when k-value is 6. When the k-value is 10, the rate of the TNR is found as 0.94, for the FOA-SVNN. It is clear from the discussion that the proposed FOA-SVNN offers a lower rate of the TNR.

4.7 Comparative discussion

The comparative analysis of the forgery detection methods is depicted in Table 1

$$\text{Performance Improvement} = \frac{per(A) - per(B)}{\text{Max(performance)}} \times 100\% \quad (37)$$

where $per(A)$ is the performance of method A and $per(B)$ is the performance of method B. The proposed method is analysed using

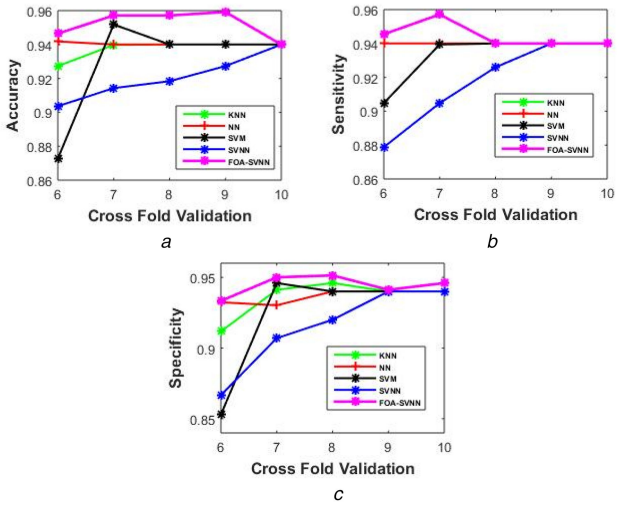

Fig. 14 *Analysis using database 2 based on cross-fold validation*
(a) Accuracy ***(b)*** Sensitivity ***(c)*** Specificity

Table 1 Comparative discussion of the forgery detection methods

	Methods	Accuracy	Specificity	Sensitivity
DSO-1 dataset	KNN [29]	0.6316	0.5625	0.8571
	NN [19]	0.7059	0.8889	0.5
	SVM [19]	0.9048	0.8462	0.94
	SVNN [28]	0.94	0.94	0.94
	Vidyadharan and Thampi [30]	—	—	0.64
	Malarvezhi *et al.* [31]	0. 8250	0. 8350	0. 8155
	proposed FOA-SVNN	**0.94**	**0.9444**	**0.94**
DSI-1 dataset	KNN [29]	0.7979	0.9467	0.5588
	NN [19]	0.8784	0.94	0.6897
	SVM [19]	0.8704	0.94	0.6701
	SVNN [28]	0.8027	0.94	0.5
	Vidyadharan and Thampi [30]	—	—	0.42
	Malarvezhi *et al.* [31]	0.8400	0.8695	0.8148
	proposed FOA-SVNN	**0.94**	**0.9583**	**0.94**

Bold values indicate the best performance.

two datasets based on sensitivity, specificity, and accuracy. For dataset 1, the accuracy of the proposed method is 32.8% higher than the accuracy of the KNN and 24.9% higher than the NN. Similarly, the accuracy of the proposed method is 3.74% and 12.23% higher than the accuracy of the existing methods, SVM and Malarvezhi *et al.*, respectively. The specificity of the proposed method is 40.43, 5.87, 10.39, and 11.58% higher than the existing methods, such as KNN, NN, SVM, and Malarvezhi *et al.*, respectively. Likewise, the sensitivity of the proposed method is 8.81, 46.8, and 31.91% higher than the sensitivity of the existing methods, such as KNN, NN, and Vidyadharan and Thampi, respectively. Similarly, for dataset 2, the proposed method has the maximum accuracy, specificity, and sensitivity than the comparative methods.

5 Conclusion

The paper focuses on a scheme for detecting the forgery for which the illumination-based texture descriptor and the FOA-SVNN-based classifier is employed. The proposed FOA-SVNN classifier aims at categorising the images as forged or non-forged images. The images are colour transformed so that to enable the easy feature extraction and the transformed image is applied to the Viola–Jones algorithm that effectively finds the face in the image. The face detected image is allowed to the feature extraction using the GWTM and the features are fed to the classifier for classification. The proposed classifier is found to be more robust and effective in identifying the forged images. The classifier is highly effective that offers better classification accuracy and it is not computationally complex. The experimentation is performed using the two datasets concerning the training percentage and the cross-fold validation. The analysis using the datasets proves that the proposed method attained an accuracy of 0.9523, sensitivity of 0.94, and specificity of 0.9583, respectively. The proposed scheme is highly accurate in detecting the presence of the forgery in the images.

6 References

[1] Carvalho, T.J., Riess, C., Angelopoulou, E., *et al.*: 'Exposing digital image forgeries by illumination color classification', *IEEE Trans. Inf. Forensics Sec.*, 2013, **8**, (9), pp. 1182–1194

[2] Hayat, K., Qazi, T.: 'Forgery detection in digital images via discrete wavelet and discrete cosine transforms', *Comput. Electr. Eng.*, 2017, **62**, pp. 448–458

[3] Mahmood, T., Mehmood, Z., Shah, M., *et al.*: 'An efficient forensic technique for exposing region duplication forgery in digital images', *Appl. Intell.*, 2017, **1**, pp. 1–11

[4] Zhao, F., Shi, W., Qin, B., *et al.*: 'Image forgery detection using segmentation and swarm intelligent algorithm', *Wuhan Univ. J. Nat. Sci.*, 2017, **22**, (2), pp. 141–148

[5] Farooq, S., Yousaf, M.H., Hussain, F.: 'A generic passive image forgery detection scheme using local binary pattern with rich models', *Comput. Electr. Eng.*, 2017, **62**, pp. 459–472

[6] Farid, H.: 'Exposing digital forgeries from JPEG ghosts', *IEEE Trans. Inf. Forensics Sec.*, 2009, **4**, (1), pp. 154–160

[7] Birajdar, G.K., Mankar, V.H.: 'Digital image forgery detection using passive techniques: a survey', *Digit. Invest. Int. J. Digit Forensic Incident Response*, 2013, **10**, (3), pp. 226–245

[8] Bhartiya, G., Jalal, A.S.: 'Forgery detection using feature-clustering in recompressed JPEG images', *Multimedia Tools Appl.*, 2017, **76**, (20), pp. 20799–20814

[9] Fadl, S.M., Semary, N.A.: 'Robust copy–move forgery revealing in digital images using polar coordinate system', *Neurocomputing*, 2017, **265**, pp. 57–65

[10] Emam, M., Han, Q., Niu, X.: 'PCET based copy–move forgery detection in images under geometric transforms', *Multimedia Tools Appl.*, 2016, **75**, (18), pp. 11513–11527

[11] Jeronymo, D.C., Borges, Y.C.C., Coelho, L.S.: 'Image forgery detection by semi-automatic wavelet soft-thresholding with error level analysis', *Expert Syst. Appl.*, 2017, **85**, pp. 348–356

[12] Gryka, M., Terry, M., Brostow, G.J.: 'Learning to remove soft shadows', *ACM TOG*, 2015, **34**, (5), pp. 1–15

[13] Schetinger, V., Iuliani, M., Piva, A., *et al.*: 'Image forgery detection confronts image composition', *Comput. Graph.*, 2017, **68**, pp. 152–163

[14] Rocha, A., Scheirer, W., Boult, T., *et al.*: 'Vision of the unseen: current trends and challenges in digital image and video forensics', *J. ACM Comput. Surv.*, 2011, **43**, (4), pp. 1–42

[15] Alkawaz, M.H., Sulong, G., Saba, T., *et al.*: 'Detection of copy–move image forgery based on discrete cosine transform', *Neural Comput. Appl.*, 2016, **34**, pp. 1–10

[16] Bharati, A., Singh, R., Vatsa, M., *et al.*: 'Detecting facial retouching using supervised deep learning', *IEEE Trans. Inf. Forensics Sec.*, 2016, **11**, (9), pp. 1903–1913

[17] Han, Q., Han, L., Wang, E., *et al.*: 'Dual watermarking for image tamper detection and self-recovery'. Proc. of the 9th Int. Conf. on Intelligent Information Hiding and Multimedia Signal Processing, Beijing, China, 2013, pp. 33–36

[18] Hu, W.C., Yang, W.H.C.D.H.C.: 'Effective image forgery detection of tampered foreground or background image based on image watermarking and alpha mattes', *Multimedia Tools Appl.*, 2016, **75**, (6), pp. 3495–3516

[19] Birajdar, G.K., Mankar, V.H.: 'Subsampling-based blind image forgery detection using support vector machine and artificial neural network classifiers', *Arab. J. Sci. Eng.*, 2018, **43**, (2), pp. 555–568

[20] Thomas, R., Rangachar, M.J.S.: 'Integrating GWTM and BAT algorithm for face recognition in low-resolution images', *Imaging Sci. J.*, 2016, **64**, (8), pp. 441–452

[21] Shen, X., Shi, Z., Chen, H.: 'Splicing image forgery detection using textural features based on the grey level co-occurrence matrices', *IET Image Process.*, 2017, **11**, (1), pp. 44–53

[22] Liu, B., Pun, C.-M., Yuan, X-C.: 'Digital image forgery detection using JPEG features and local noise discrepancies', *Sci. World J.*, 2014, **2014**, pp. 1–12

[23] Peng, B., Wang, W., Dong, J., *et al.*: 'Optimized 3D lighting environment estimation for image forgery detection', *IEEE Trans. Inf. Forensics Sec.*, 2017, **12**, (2), pp. 479–494

[24] Felzenszwalb, P.F., Huttenlocher, D.P.: 'Efficient graph-based image segmentation', *Int. J. Vis. Comput.*, 2004, **59**, (2), pp. 167–181

[25] Buchsbaum, G.: 'A spatial processor model for color perception', *J. Franklin Inst.*, 1980, **310**, (1), pp. 1–26

[26] Mathur, M.K., Bhati, P.: 'Face objects detection in still images using Viola–Jones algorithm through MATLAB tools', *Int. J. Innov. Res. Comput. Commun. Eng.*, 2017, **5**, (2), pp. 2468–2476

[27] Pan, W.: 'A new fruitfly optimization algorithm: taking the financial distress model as an example', *Knowl.-Based Syst.*, 2012, **26**, pp. 69–74
[28] Ludwig, O., Nunes, U., Araujo, R.: 'Eigen value decay: a new method for neural network regularization', *Neurocomputing*, 2014, **124**, pp. 33–42
[29] Baby, L., Jose, A.: 'Digital image forgery detection based on GLCM and HOG features, international journal of advanced research in electrical', *Electron. Instrum. Eng.*, 2014, **3**, (5), pp. 426–430
[30] Vidyadharan, D.S., Thampi, S.M.: 'Detecting spliced face in a group photo using PCA'. Proc. of 7th Int. Conf. on Soft Computing and Pattern Recognition (SoCPaR), Fukuoka, Japan, 2015, pp. 175–180
[31] Malarvezhi, P., Prashanth, M.S., Abineash, R.R.*, et al.*: 'Illumination map based image splicing detection', *Int. J. Control Theory Appl.*, 2017, **10**, (30), pp. 177–185