PeerJ Computer Science

Deep learning model for deep fake face recognition and detection

Suganthi ST[1], Mohamed Uvaze Ahamed Ayoobkhan[2], Krishna Kumar V[3], Nebojsa Bacanin[4], Venkatachalam K[5], Hubálovský Štěpán[5] and Trojovský Pavel[6]

[1] Department of Computer Engineering, Lebanese French University,Iraq., Erbil, Iraq
[2] Computing Department, Westminster International University in Tashkent, Tashkent, Uzbekistan
[3] Department of Computer Science Engineering, Sri Ramakrishna Engineering College, Coimbatore, India
[4] Department of Computing, Singidunum University, Belgrade, Serbia
[5] Department of Applied Cybernetics, Faculty of Science, University of Hradec Kralove, Hradec Kralove, Czech Republic
[6] Department of Mathematics, Faculty of Science, University of Hradec Kralove, Hradec Kralove, Czech Republic

ABSTRACT

Deep Learning is an effective technique and used in various fields of natural language processing, computer vision, image processing and machine vision. Deep fakes uses deep learning technique to synthesis and manipulate image of a person in which human beings cannot distinguish the fake one. By using generative adversarial neural networks (GAN) deep fakes are generated which may threaten the public. Detecting deep fake image content plays a vital role. Many research works have been done in detection of deep fakes in image manipulation. The main issues in the existing techniques are inaccurate, consumption time is high. In this work we implement detecting of deep fake face image analysis using deep learning technique of fisherface using Local Binary Pattern Histogram (FF-LBPH). Fisherface algorithm is used to recognize the face by reduction of the dimension in the face space using LBPH. Then apply DBN with RBM for deep fake detection classifier. The public data sets used in this work are FFHQ, 100K-Faces DFFD, CASIA-WebFace.

Subjects Algorithms and Analysis of Algorithms, Artificial Intelligence, Brain-Computer Interface, Data Mining and Machine Learning, Distributed and Parallel Computing
Keywords Deep fake, Fisherface, LBPH, DBN, RBM, Deep learning

Submitted 6 December 2021
Accepted 19 January 2022
Published 22 February 2022

Corresponding author
Suganthi ST, suganthi.sb@gmail.com

Academic editor
Ali Kashif Bashir

Additional Information and Declarations can be found on page 17

DOI **10.7717/peerj-cs.881**

INTRODUCTION

Digital manipulation of the face images include facial information of fake images using deep fake approaches (*Korshunov & Marcel, 2018*). In recent years, deepfake approach has become a popular technique in detecting fake images recently (*Citron, 2019*; *Cellan-Jones, 2019*). It is implemented by using deep learning technique in order to create fake images by swapping the face of one person by the face of another. In the year 2017, it was termed as Reddit user and it is commonly known as ''deep fakes'' using deep adversarial models (*i.e.*) Generative Adversarial Networks (GAN) and similarly, it transform the celebrity faces into porn videos (*Maras & Alexandrou, 2019*). The main issues in the fake pornography are including fake news in the content, financial fraud and hoaxes. At the same time, the advantages of deep fake is in the fields such as virtual reality, editing film and

production. The chief working concepts of deep fakes are merging, replacing, combining and superimposing images using deep learning and machine learning techniques so as to create fake digital images or videos (*Maras & Alexandrou, 2019*).

Many software apps/tools are available through which deep fake images are created without a programming knowledge and technical side background information. Usually the profile pictures from the social media are taken and fake images or videos are developed with a help of the expert. Security enhancement in the detection of face swap and the accuracy are very low. To overcome these issues, this paper proposes a new strategy for detecting the deep fake facial images using the fisher face with an LBPH approach. In the digital image manipulation, techniques are applied in many fields which give more misinformation in the society. The scenarios such as creating fake news, providing false information in the political elections, create security threats (*Allcott & Gentzkow, 2017*; *Lazer et al., 2018*).

CNN based methods like XceptionNet, Meso Inception-Net, ResNet are used in the field of detecting deep fakes which include detection of visual artifacts, inconsistency in color during the time of performing blend operations in the image analysis. The process include identifying the forgery of face X-ray by blending the boundary forged image in the CNN model and classifying the loss in the detection of face X-ray (*Li et al., 2020*; *Li & Lyu, 2018*; *Afchar et al., 2018*; *Dang et al., 2020*). In the media, articles use the biometric technology for the detection of deep fakes. In order to detect the difference between real and fake images in the field of ocular biometric, CNN methods such as Squeeze Net, DenseNet, ResNet and light CNN are used (*Nguyen & Derakhshani, 2020*). The main contributions of this research work are the following:

1. A new hybrid high-performance deep fake face detection method is used based on the analysis of the Fisher face algorithm (LBHH) with dimensional reduction in features of the face image.
2. To detect the fake and real image using deepfake detection classifier based on DBN with the RBM technique.

The paper has been organized as follows: section 2 describes about the review of literature, section 3 introduces deep fake detection using FF-LBPH-DBN, section 4 explains about the experimental results and section 5 concludes the paper with future directions.

RELATED WORKS

Deep fakes in the face manipulation are the frightening thing to distort the original facts in the digital images. In the advancement of technologies, deep fake detection of algorithms are necessary to be used these days in verifying the content of digital manipulation information. By using deep learning algorithms such as Generative Adversarial Neural Networks (GANs) are based on the concept of auto encoders and decoders for the implementation of detecting the fake images or videos (*Yadav & Salmani, 2019*). Deep fakes which include swapping of face images are carried out without the knowledge of the celebrities. It is also used to misrepresent the face images of the politicians. At first, swapping of face image was done in the photo of Abraham Lincoln (*Badale et al., 2018*). *Yang, Li & Lyu (2019)* have proposed

a model to detect deep fake using head poses inconsistency. By using that model, the faces for various persons were created without modifying the original face expressions. *Jagdale & Shah (2019)* paper proposed an algorithm of NA-VSR for super resolution. The concept of the algorithm is that it reads the video and converts into frame by frame (*Maheswaran et al., 2017*). Then, the median filter is applied to remove the unnecessary noise in the video. By the use of bicubic interpolation technique, the density of the pixel in the image gets increased. Bicubic transformation is applied for the enhancement of the image. *Yadav & Salmani (2019)* have described the working principle of the deep fake techniques along with swapping of face images in a high precision value (*Maheswaran et al., 2018*). Generative Adversarial Neural Networks (GANs) contain two neural networks; one is generator and the other is discriminator. In the generator neural networks, the fake images are created from the given data set. At the same time, discriminator neural networks are used to evaluate the images which are synthesized by the generator and check its authenticity. The important problems of deep fake are so harmful due to defamation of individual character and assassination and spreading fake news in the society.

There are many such issues in the existing approaches in terms of inefficiency in detecting the deep fake images, high error rate, high consumption time also high and inaccuracy in accessing the data. This work FF-LBPH-DBN focuses mainly on the minimization of computation and the application for various metrological parameters in an efficient way. Table 1 shows the survey based on detection of the fake images (*Vivek et al., 2018*).

METHODOLOGY

The proposed face recognition and fake detection is based on the deep learning technique of the fisherface using the Local Binary Pattern Histogram (FF-LBPH). The accurate detection of deep fake image system consists of four phases such as (i) pre-processing, (ii) dimensionality reduction of image (iii) feature extraction and (iv) classification. This architecture of proposed work diagram is given in Fig. 1.

Figure 1 shows the pre-processing phase which includes resizing of images, removal of noise and normalization. For the improvement of feature extraction and classification process, the dimensionality reduction of face image is used by the fisherface algorithm (LBPH).

Pre-Processing

For classifying the deep fake image pre-processing is needed which enhancing the image for further processing. The steps involved in the phase of pre-processing are given in Fig. 2.

Figure 2 shows the three stages of pre-processing namely, resizing of image, removal of noise and normalization.

Resize image

In the data set, all the images were in various sizes and the processing of various size data could not provide accurate result. All the images were resized as 256 $\times$ 256 and it was

Table 1 Survey of deepfake algorithms.

Author name	Name of the method	Classifier	Data set
Guarnera, Giudice & Battiato (2020)	Pipeline Features uisng GAN	k-NN, SVM, LDA	Own (AttGAN, GDWCT, StarGAN, StyleGAN, StyleGAN2)
Neves et al. (2020)	Deep Learning	CNN	100K-Faces (StyleGAN)iFakeFaceDB
Dang et al. (2020)	Deep Learning	fusion of CNN with Attention Mechanism	DFFD (ProGAN, StyleGAN)
Hulzebosch, Ibrahimi & Worring (2020)	Deep Learning	CNN, AE	StarGAN, Glow, ProGAN, StyleGAN
Chen et al. (2020)	Deep Learning	CNN, LSTM	UADFV, Celeb-DF
Ranjan, Patil & Kazi (2020)	Deep Learning	CNN, LSTM	FaceForensics++, Celeb-DF, DeepFake Detection Challenge
Wang et al. (2019)	GAN-Pipeline	SVM	(InterFaceGAN, StyleGAN)
Nataraj et al. (2019)	Steganalysis	CNN	100K-Faces (StyleGAN)
Yu, Davis & Fritz (2018)	Deep Learning	CNN	(ProGAN, SNGAN, CramerGAN, MMDGAN)
Marra et al. (2019)	Deep Learning	CNN	(CycleGAN, ProGAN,Glow, StarGAN, StyleGAN)
Mahendhiran & Kannimuthu (2018)	Deep Learning	KNN, navie bayes, Neura network, random forest	Multimodal sentimental prediction
Arunkumar & Kannimuthu (2020)	Evolutionary model	Bird eye view methods	Data analytics

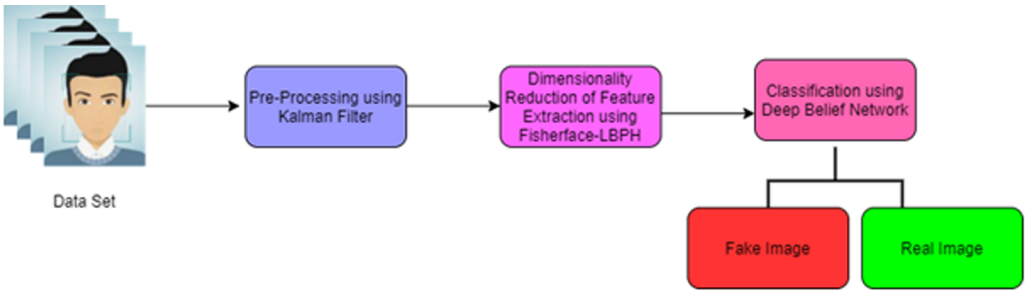

Figure 1 Architecture of proposed work.
Full-size DOI: 10.7717/peerjcs.881/fig-1

used for further processing. For resizing the input image downsampling and upsampling methods were employed.

Removal of noise

In order to improve the efficiency in the classification of deep fake images, the noise was removed from raw input face image by using Kalman filter. Generally, it is a recursive mathematical model and it consists of two different processes; the prediction process and the update process. In the prediction process, priori system state is estimated from the

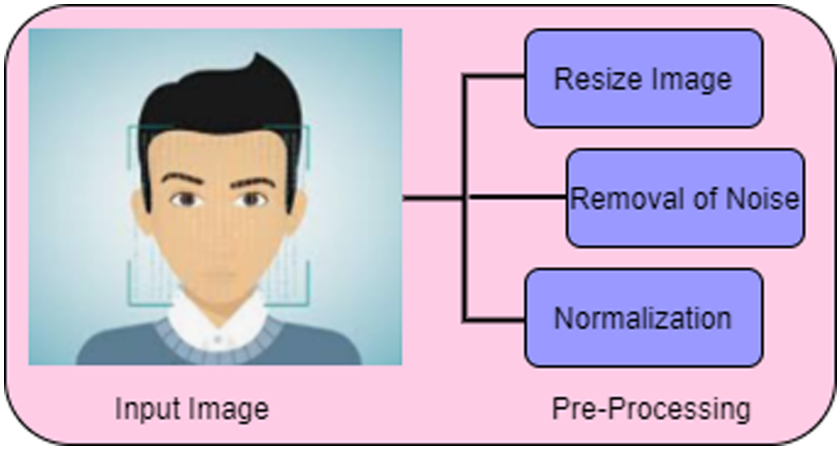

Figure 2 **Pre-processing.**

Full-size DOI: 10.7717/peerjcs.881/fig-2

Table 2 **Filtering loops.**

Prediction process	Update process
The priori estimate is calculated in the prediction process by using Eqn. $\hat{X}_n^- = T_n.\hat{X}_{n-1}^+$	Kalman Gain matrix is represented using Eqn $KG_n = P_n^-.O_n^t.(O_n.P_n^-.H_n^t + OUN_n)$
The covariance matrix is calculated using Eqn $C_n^- = T_n.P_{n-1}^+.T_n^t + PPN_n$	The posteriori estimate is evaluated by using the Eqn $Z_n : \hat{X}_n^+ = \hat{X}_n^- + KG_n.(Z_n - O_n.\hat{X}_n^-)$
	Posteriori estimate covariance matrix is calculated as in Eqn $P_n^+ = (I - KG_n.O_n).P_n^-$

previous state. And in the update process, the posteriority state is determined with the correction of priori state. The initial estimate state x_0^- is repeated until the filtering process ends (*Kalman, 1960*; *Arasaratnam, Haykin & Hurd, 2010*). The Kalman filtering looping state is shown in Table 2.

The parameters of Kalman filter are necessary to tune with covariance matrices of noises such as PPN, OUN and P_0^+. These covariance matrices are used to predict the weights. The noise of this filter has the zero multivariate Gaussian distribution of these covariance matrices. The covariance matrix of the sample vector $X = [X_1, X_2, ..X_n]^T$ is represented in Equation 1.

$$\sum = \begin{pmatrix} \sum_{1,1} & \cdots & \sum_{1,n} \\ \vdots & \ddots & \vdots \\ \sum_{n,1} & \cdots & \sum_{n,n} \end{pmatrix} \tag{1}$$

where $\sum_{i,j} = cov\left(X_i, X_j\right) = E\left(X_i - \eta_i\right)\left(X_j - \eta_j\right)$, $\eta_i = E[X_i]$, and where E is the expectation operator. The filter tuning is address with two approaches such as static and dynamic. The static tuning tunes the filter before the usage of it with the techniques such as autocovariance least squares (ALS). And the dynamic tuning tunes the filter while it is

operating with self-tuning. Moreover, it uses the method called Artificial Neural Network. Once the data are pre-processed using the Kalman filter, the pre-processed data are then given as input to the feature selection phase in order to select the relevant features for classification.

Normalization

For the enhancement, the contrast of image was used by using normalization. It was carried out based on pixel intensity value. The normalization process of this proposed work had used RGB pixel compensation method. It was based on the adaptive illumination of compensation dependent on the black pixel with histogram equalization.

Dimensionality reduction using proposed fisherface-LBPH

Dimensionality reduction is an important step to reduce the dimension of the input image into low dimensional space. The proposed work was based on fusion of fisherface with Local Binary Pattern Histogram (FF-LBPH) which was utilized in the reduction of face space dimension. The fusion of fisherface with Linear Binary Pattern Histogram (LBPH) was implemented in the proposed study.

Fisherface

The particular technique is based on Fisher's Linear Discriminant Analysis (FLDA). The main advantage of the fisherface algorithm is faster in execution when compared to the eigenface technique. It is prominent for low error rates and also it works efficiently in various illuminations with different facial expressions. Steps involved in the fisherface algorithm are given below,

---

Algorithm 1: Dimensionality reduction in feature extraction Fisher face (Proposed)

Input: Face Image from the data set

Output: Dimensionality reduction in feature extraction

Step 1: Assume that size of the square face image with $height = width = N$ and img is the number of images in the database.

Step 2: Select sample images form the database $\left\{\overrightarrow{a}, \overrightarrow{b}, \ldots, \overrightarrow{e}\right\}$ and class scatter $c = \{x_1, x_2, \ldots, x_n\}$

$$\text{face image 1} = \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{N^2} \end{Bmatrix}; \text{face image 2} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_{N^2} \end{Bmatrix}; \text{face image 3} = \begin{Bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_{N^2} \end{Bmatrix};$$

$$\text{face image 4} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N^2} \end{Bmatrix}; \text{face image 5} = \begin{Bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_{N^2} \end{Bmatrix}; \text{face image 6} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{N^2} \end{Bmatrix}$$

Step 3: Calculate the average of all faces by using:

$$\overrightarrow{m} = \frac{1}{img}\begin{bmatrix} a_1+b_1+\cdots+f_1 \\ a_2+b_2+\cdots+f_2 \\ \vdots \quad \vdots \quad \vdots \\ a_{N^2}+b_{N^2}+\cdots+f_{N^2} \end{bmatrix} \tag{2}$$

where $img = 6$

Step 4: Calculate the average face of each image in the data set $\overrightarrow{img}_1 = \frac{1}{2}\begin{bmatrix} a_1+b_1 \\ a_2+b_2 \\ \vdots \ \vdots \\ a_{N^2}+b_{N^2} \end{bmatrix}; \overrightarrow{img}_2 = \frac{1}{2}\begin{bmatrix} c_1+d_1 \\ c_2+d_2 \\ \vdots \ \vdots \\ c_{N^2}+d_{N^2} \end{bmatrix}; \overrightarrow{img}_3 = \frac{1}{2}\begin{bmatrix} e_1+f_1 \\ e_2+f_2 \\ \vdots \ \vdots \\ e_{N^2}+f_{N^2} \end{bmatrix}$ **Step 5:** Subtract the average face of each image from the training images

$$\overrightarrow{img}_{1m} = \begin{bmatrix} a_1-img_{11} \\ a_2-img_{12} \\ \vdots \quad \vdots \\ a_{N^2}-img_{1N^2} \end{bmatrix}; \overrightarrow{img}_{2m} = \begin{bmatrix} b_1-img_{11} \\ b_2-img_{12} \\ \vdots \quad \vdots \\ b_{N^2}-img_{1N^2} \end{bmatrix}; \overrightarrow{img}_{3m} = \begin{bmatrix} c_1-img_{21} \\ c_2-img_{22} \\ \vdots \quad \vdots \\ c_{N^2}-img_{2N^2} \end{bmatrix}$$

$$\overrightarrow{img}_{4m} = \begin{bmatrix} d_1-img_{21} \\ d_2-img_{22} \\ \vdots \quad \vdots \\ d_{N^2}-img_{2N^2} \end{bmatrix}; \overrightarrow{img}_{5m} = \begin{bmatrix} e_1-img_{31} \\ e_2-img_{32} \\ \vdots \quad \vdots \\ e_{N^2}-img_{3N^2} \end{bmatrix}$$

$$; \overrightarrow{img}_{6m} = \begin{bmatrix} f_1-img_{41} \\ f_2-img_{42} \\ \vdots \quad \vdots \\ f_{N^2}-img_{4N^2} \end{bmatrix} \tag{3}$$

Step 6: Create scatter matrix $sm_1,\ sm_2,\ sm_3, sm_4$

$$sm_1 = \left(\overrightarrow{img}_{1m}\overrightarrow{img}_{1m}^{T} + \overrightarrow{img}_{2m}\overrightarrow{img}_{2m}^{T}\right) \tag{4}$$

$$sm_2 = \left(\overrightarrow{img}_{3m}\overrightarrow{img}_{3m}^{T} + \overrightarrow{img}_{4m}\overrightarrow{img}_{4m}^{T}\right) \tag{5}$$

$$sm_3 = \left(\overrightarrow{img}_{5m}\overrightarrow{img}_{5,m}^{T} + \overrightarrow{img}_{6m}\overrightarrow{img}_{6m}^{T}\right) \tag{6}$$

Step 7: Construct a scatter matrix within the class $s_{mw} = sm_1 + sm_2 + sm_3$.

Step 8: Construct the scatter matrix between class

$$s_{mb} = 2\left(\overrightarrow{img}_1 - \overrightarrow{m}\right)\left(\overrightarrow{img}_1 - \overrightarrow{m}\right)^T + 2\left(\overrightarrow{img}_2 - \overrightarrow{m}\right)\left(\overrightarrow{img}_2 - \overrightarrow{m}\right)^T$$

$$+2\left(\overrightarrow{img}_3 - \overrightarrow{m}\right)\left(\overrightarrow{img}_3 - \overrightarrow{m}\right)^T \tag{7}$$

Step 9: Compute the vector vec_{img} and the columns of vec_{img} contain eigen vector values for $s_{mw}^{-1}s_{mb}$. Here s_{mw} is minimized; s_{mb} is maximized by using:

$$vec_{img} = \left| \frac{vec_{img}^{T} s_{mb}}{vec_{img}^{T} s_{mw}} \right| \tag{8}$$

Generally, it can be defined by the decomposition of eigen value and it is represented as:

$$s_{mb} vec_{img} = s_{mw} vec_{img} \Lambda \tag{9}$$

Where, vec_{img} is eigen vector matrix and Λ are eigen values in the diagonal matrix. Eigen vectors vec_{img} are associated with eigen values of non-zero which are the fisherfaces.

Step 10: Normalization of the Equation

Step 11: Evaluate the weight for training image in the dataset in the normalized fisherface.

Step 12: Extracting features using dimensionality reduction of features so as to obtain face identification.

Algorithm 1 is an improvement version of eigen faces which include Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). In order to get a sub-space and to maximize the variability within classes and between the classes of scatter matrix.

Linear Binary Pattern Histogram (LBPH)

LBPH was used to recognize the facial images in the database. Extracting the features of face image and using binary operator, it recognized the image with less computational time complexity. Algorithm 2 describes LBPH.

Algorithm 2: LBPH

Input: Input Face Image

Output: LBP pixel value

Step 1: Split the face image into $n \times n$ (i.e) 8×8 which contains 64 parts or regions.

Step 2: Extraction of histogram values from each 64 sub-regions of face image using

$$hist_{i,j} = \sum_{x,y} Img\left\{f_{img}(x,y) = i\right\} Img\left\{(x,y) \in reg_j\right\} \tag{10}$$

$where,\ \ i = 0\ to\ n-1\ \&\ j = 0\ to\ m-1$

m is the total number of sub-regions

n is the total number of class labels created by LBP operator.

Step 3: Apply Local binary operator in every sub-region and it is applied in 8×8 window size using

$LBP(i,j) = \sum_{pix=0}^{pi-1} 2^{pi}\ su\left(img_{npix} - img_{c}\right)$ where, i,j is the centre pixel value of intensity img_c, img_{npix} is the neighbour pixel value of intensity,

su is the sub region of the image.

Step 4: Select pixel value of median as threshold value and compare it with neighbourhood pixel value of image 8×8 window size.

$$su(x) = \begin{Bmatrix} 1, & x \geq 0 \\ 0, & x < 0 \end{Bmatrix} \tag{11}$$

If the neighbour pixel value is greater than or equal to the middle pixel value as 1, the value is set value as 0.

Step 5: Combine all the neighbour pixel values to form 8-bit binary number and convert it into decimal number and it is called as LBP pixel value. (range from 0-255).

In Algorithm 2, LBPH is the fusion of Local Binary Patterns (LBP) technique with Histograms of Oriented Gradients (HOG) descriptor. It is a simple and powerful method to extract the features and labelling the pixels in the face image.

Dimensionality Reduction in feature extraction using proposed fisherface-LBPH

By reducing the dimension of face image and extracting the features using fusion of the fisherface algorithm with LBPH, the face image got recognized. The steps involved (FF-LBPH) are given below:

Algorithm 3: Dimensionality reduction in feature extraction of fisherface-LBPH (Proposed)

Input: Face Image from the data set

Output: Recognizing the face image

Step 1: Read face image of size $m \times m$ from the dataset and stored in form of column vector values.

Step 2: Normalize the input face training image and calculate the value of various matrix by subtracting the average value from the training image.

Step 3: Evaluate Algorithm 1 and extract the relevant features of face image.

Step 4: call algorithm 2

Step 5: Count the similar LBP pixel value in all sub region of face image.

Step 6: Combine all histogram value into single histogram value and it is stored as vector value for features of the face image.

Step 7: To recognize the similar images in the testing data set by performing the match process of testing image and calculating the minimum distance between original image and testing image using Euclidean distance:

$$dist\,(a,b) = \sum_{i=1}^{n} \left|\left| \lfloor histimg_1 - histimg2 \rfloor \right|\right| \tag{12}$$

Step 8: Recognize the matching images.

In Algorithm 3, fusion of fisherface with LBPH is implemented. At first, it takes the image in same height and width. It extracts the features using the concept of principal components which differentiate one face image of individuality from the another. Therefore, each

and every feature of the image cannot dominate the another. In order to obtain the characteristics of the face image features by reducing the face image space dimensions using fisher Linear discriminant (FDL) technique. After obtaining the features of face image,LBPH is applied in order to get a fine tune classification of deepfake face image. In method of LBPH, if the neighbourhood value is greater than the threshold (median value), it is taken as 1 else 0. Considering this value as a binary format, it is converted to a decimal format. Hence, this decimal format is called as LBP value. After generating the LBP value, the histogram of subregion is evaluated and the similar LBP values in the subregion of face image are counted. Then, all the histograms of subregion are merged to form a single histogram which is called as feature vector of the face image. By comparing the histogram of test face image with all images in the dataset, the closest histogram value of face image is recognized.

Deepfake detection using classifier

Background of DeepFake

In the synthetic media, deepfakes are replacing the existing face image with image of someone else. It uses Generative Adversarial Networks (GANs) for manipulating the faces. The facial manipulation contains three phases namely face synthesis, face swap and facial attributes and expressions.

Face Synthesis. Using GAN in this phase replaces the real face with the fake image. The best approach used in this phase is StyleGAN. In the StyleGAN unsupervised training process is implemented and it generates the images with variations such as hair, freckles, etc. It also enables the synthetic controls of the image.

Face swap. In the face manipulation, face swap is one of the popular techniques. It is used to detect the image or video of a person fake or real after swapping its face in the image. The most popular database which contains real and fake videos are FaceForensics++. From the database, the fakevideos are used with the help of FaceSwap computer graphics concept and the other deep learning techniques such as DeepFakeFaceSwap.

Facial attributes and expressions. Facial attribute modifications include color of skin, hair, age, gender. Similarly, change of face expressions include sad, happy, anger and so on. These are called as manipulation of facial attributes and its expressions. Moreover, the most popular mobile app is FaceApp. It uses StarGAN methos for performing the image-to image translation (*Tolosana et al., 2020*).

Deepfake detection using classifier of DBN-RBM

Deepfake is a technique which uses the Generative Adversarial Networks (GANs) for generating fake images. Deepfake detection is based on the classifier algorithm Deep Belief Network (DBN) which is used to classify fake images from authentic image. DBN technique consists of three layers such as input, hidden and output layers. In addition to that, deep learning network contains stacked hidden layers and it is the extension of the neural network. DBN consists of one visible layer and multiple hidden layers. Transmission of input face image through visible layer to hidden layer is activated through sigmoid function

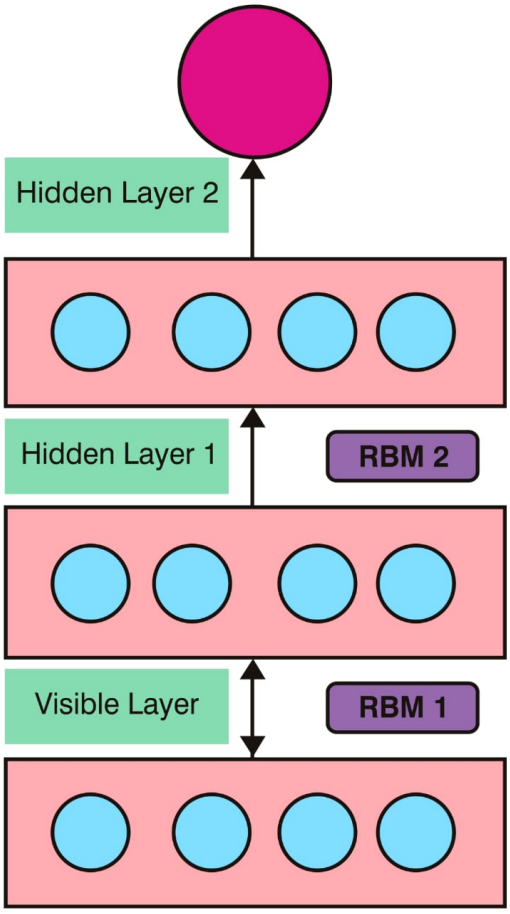

Figure 3 **DBN with RBM.**

Full-size DOI: 10.7717/peerjcs.881/fig-3

based on the RBM learning rule (*Hinton, Osindero & Teh, 2006*). It is based on Restricted Boltzmann Machine (RBM) and DBN is acted with RBM which communicates with the previous layer and the subsequent layers in the DBN network. The architecture of DBN with RBM is shown in Fig. 3.

Figure 3 shows the architecture of DBN that consists of two stacked RBMs. RBM1 consists of visible layer and hidden layer 1, RBM2 consists of hidden layer 1 and hidden layer 2. In this architecture, the input face image is trained in the DBN based RBM classifier with the learning rule. The parameters which are used in the DBN architecture contain weight values between visible layer and hidden layer, value of bias and neuron states. Sigmoid function is applied for the transformation of neuron values from previous layer to the next layer using:

$$P\left(sigmo_i = 1\right) = \frac{1}{1+\exp(-b_i - \sum_j sigmo_j w_{ij})}. \tag{13}$$

Bias and weight of all neurons are initialized in the RBM layer. In the training, the input face image consists of positive and negative phase. In the positive phase, input data is transformed from visible layer to hidden layer and negative phase transforms the input data from hidden layer to visible layer. In the positive and negative phases, individual activation function is calculated by using Eqn and that are defined as.

$$P(v_i = 1|h) = sigmo(-b_i - \sum_j h_j w_{ij}) \tag{14}$$

$$P(h_i = 1|v) = sigmo(-c_i - \sum_j h_j w_{ij}) \tag{15}$$

where, v_i isthe visible layer; h_i is the hidden layer and w_{ij} is the weight value.

This process is repeated and updated the weight value in the DBN architecture, until the maximum number of epochs is reached. The training process continued and the parametric values are optimized using:

$$update(w_{ij} + \frac{\eta}{2} \times (positive(E_{ij}) - negative(E_{ij}))) \tag{16}$$

where,

$positive(E_{ij})$-Positive statistics of edge $E_{ij} = p(h_j = 1|v)$
$negative(E_{ij})$-Positive statistics of edge $E_{ij} = p(v_j = 1|h)$
η- learning rate

Using the above procedure, RBM is trained and the same process is repeated until all RBM get trained. By using the proposed work of dimensionality reduction in feature extraction, fisherface-LBPH was used for feature extraction with DBN-RBM classifier and it was used to recognize and differentiate the fake image and the real image.

RESULT & DISCUSSIONS

The proposed deep learning dimensionality reduction in feature extraction fisherface-LBPH evaluated the real and fake images in the public dataset. The public datasets used for deepfake detection were FFHQ, 100K-Faces, DFFD, CASIA-WebFace .

Data set description

Flickr-Faces-HQ, FFHQ

Flickr-Faces-HQ, FFHQ dataset contains a group of 70,000 face images with a high-quality resolution generated by generative adversarial networks (GAN).

100K-Faces

100K-Faces dataset contains 100,000 unique human face images generated using StyleGAN

Fake face dataset (DFFD)

DFFD dataset contains 100,000 and 200,000 fake images generated by ProGAN and StyleGAN. The dataset includes approximately 47.7 percent of male images, 52.3 percent of female images, and most of the sample images are in the range of age from 21 to 50 years old.

CASIA-WebFace

CASIA-WebFace database contains 10,000 subjects and 500,000 images. These images are crawled from IMDB website which has 10,575 of a well-known actors and actresses of IMDB.

Performance metric measures

Performance metric measures such as accuracy and error detection rate are evaluated. To determine the performance of the proposed algorithm, it is compared with the existing approaches such as Supprot Vector Machine (SVM), LDA , KNN and Convolution Neural Network (CNN) . In order to evaluate the performance metric measures, accuracy, sensitivity, specificity, error rate of the Root Mean Square Error (RMSE), Signal-to-Noise-Ratio (SNR), Peak Signal-to-Noise-Ratio (PSNR), and Mean Absolute Error (MAE) are utilized.

Accuracy

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} X100 \tag{17}$$

Sensitivity

$$Sensitivity = \frac{TP}{TP + FN} X100 \tag{18}$$

Specificity

It is used to evaluate the rate between True Negative (TN) and True Positive (TP)

$$Specificty = \frac{TN}{TN + FP} X100 \tag{19}$$

The error rate is given below:

$$PSNR = 20\log_{10}\left(\frac{255^2}{MAE}\right) \tag{20}$$

$$MAE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|X\left(i,j\right) - Y\left(i,j\right)\right| \tag{21}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_i - \widehat{X}_i\right)^2} \tag{22}$$

$$SNR\,(db) = 20log\left(\frac{V_{RMS(Signal)}}{V_{RMS(Noise)}}\right) \tag{23}$$

Table 3 Performance comparison of proposed methods with different datasets in terms of accuracy.

Methods	Datasets			
	FFHQ	100K-Faces	DFFD	CASIA-WebFace
SVM	82.5	70.12	84.43	85.25
LDA	86.32	78.11	88.32	84.52
KNN	88.15	80.21	87.01	91.75
CNN	89.23	82.45	88.55	86.12
Proposed FF-LBPH-DBN	94.92	95.55	97.82	98.82

Table 4 Performance comparison of proposed methods with different datasets in terms of error detection rate.

Methods	Datasets			
	FFHQ	100K-Faces	DFFD	CASIA-WebFace
SVM	15.37	26.81	13.97	12.48
LDA	14.25	24.02	13.75	12.82
KNN	12.33	16.56	12.78	10.19
CNN	12.23	15.25	12.78	10.25
Proposed FF-LBPH-DBN	9.12	11.25	9.04	7.06

Table 5 Sensitivity and specificity in different data sets.

Data Sets	Sensitivity %					Specificity %				
	SVM	LDA	KNN	CNN	FF-LBPH-DBN	SVM	LDA	KNN	CNN	FF-LBPH-DBN
FFHQ	85.3	796	83.9	82.4	89.67	82.75	85.7	81.45	86.78	91.22
100K-Faces	84.2	81	83.8	81.8	86.88	83.75	85.9	89.54	86.89	92.45
DFFD	82.9	86.3	82.7	80.9	88.9	86.75	86.9	88.45	84.56	93.76
CASIA-WebFace	89.2	81.2	89	87	91.35	85.78	88.4	85.12	87.91	94.35

Table 2 shows that accuracy rate of proposed work that is compared with the different data set.

Table 3 shows the accuracy of various algorithm with the proposed work and it is implemented in various public available data set such as FFHQ, 100K-Faces, DFFD and CASIA-WebFace. The accuracy rate of proposed work FF-LBPH-DBN was high (98.82%) in the dataset of CASIA-WebFace image dataset. The next position in terms of accuracy rate is 97.82% for DFFD dataset. Table 4 shows error detection rate of proposed work in various data set.

Table 4, shows the error rate of various algorithm in different data sets. The proposed work FF-LBPH-DBN got minimum error rate of 7.06 in the data set CASIA-WebFace data set. Table 5 shows the sensitivity, specificity performance comparison using various techniques namely SVM, LDA, KNN, CNN. The proposed work FF-LBPH-DBN with various datasets of FFHQ, 100K-Faces, DFFD, CASIA-WebFace were provided for better understanding.

Table 6 EER and AUC on deepfake detection methods.

Data Sets	Deepfake		Face swap		Face Synthesis	
	AUC	EER	AUC	EER	AUC	EER
FFHQ	0.948	13.33	0.918	13.65	0.726	16.49
100K-Faces	0.975	9.57	0.954	10.76	0.772	19.51
DFFD	0.969	12.45	0.944	12.78	0.714	15.67
CASIA-WebFace	0.978	7.21	0.986	9.56	0.788	12.32

Table 5 shows the sensitivity, specificity that provide best performance for FF-LBPH-DBN algorithm compared to the existing algorithms and various data sets of FFHQ, 100K-Faces, DFFD, CASIA-WebFace. Whereas, FF-LBPH-DBN of proposed work got sensitivity score as 89.67% in FFHQ data set, 86.88% in 100K-Faces data set, 88.9% in DFFD data set and 91.35% in CASIA-WebFace data set. Similarly for Specificity of proposed work FF-LBPH-DBN got 91.22% in FFHQ data set, 92.45% in 100K-Faces data set, 93.76% in DFFD data set and 94.35% in CASIA-WebFace data set. The deepfake detection of face image in the aspects of Equal Error Rate (EER) and AUC was done on the datasets of FFHQ, 100K-Faces, DFFD, CASIA-WebFace. These datasets were used in both training and testing process by deepfake detection classifier methods namely, deepfake, face swap, face synthesis with the proposed work of FF-LBPH-DBN model. Table 4 shows the performance of deepfake detection in the aspects of Equal Error Rate (EER) and AUC on various datasets.

Table 6 shows the exact recognition of real and fake images for deepfake, faceswap and face synthesis methods. In the proposed work, the dataset of CASIA-WebFace had attained better performance in the methods of deepfake with 0.978 in AUC and 7.21 in EER, faceswap with 0.986 in AUC and 9.56 in EER and facesynthesis with 0.788 in AUC and 12.32 in EER. Figure 4 shows the error rate value that is calculated based on its accuracy using the Eq22–Eq25.

From Fig. 4, it is observed that PSNR value must be increased and MAE value must be decreased for the best detection of fake face image and real face image. The proposed approach provided better error rate value with the base of accuracy. In the proposed work the value of PSNR was increase and the value of MAE got decreased when compared to the other existing techniques. Figure 5 shows the computation time for various algorithms.

In Fig. 5, it is revealed that the proposed algorithm of FF-LBPH-DBN needs less computation time when compared to the other existing algorithms. The analysis of training and testing of dataset in the face image had produced validation face input data in the terms of accuracy and loss metric information in the deepfake face image dataset. The proposed work of FF-LBPH-DBN model in epochs is shown in Fig. 6.

It is observed in Fig. 6 that the proposed methods FFHQ, 100K-Faces, DFFD, CASIA-WebFace of dataset provide less validation loss and good validation accuracy for FF-LBPH-DBN model.

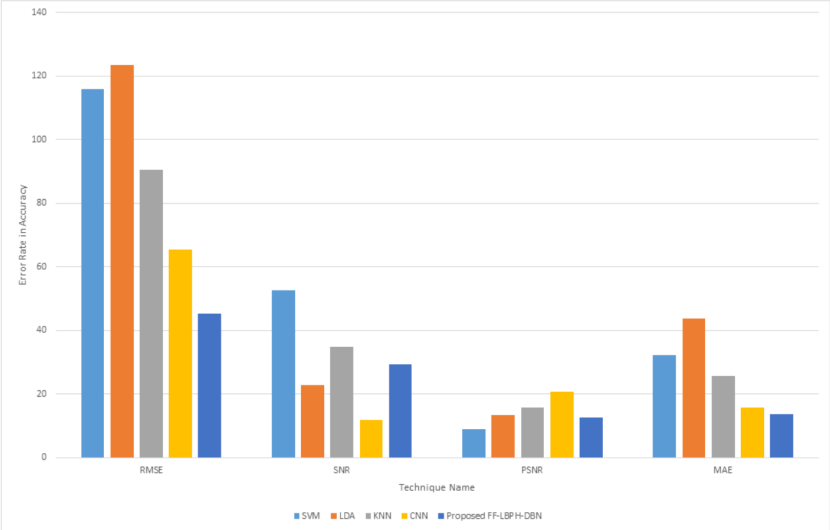

Figure 4 **Error rate in accuracy.**

Full-size DOI: 10.7717/peerjcs.881/fig-4

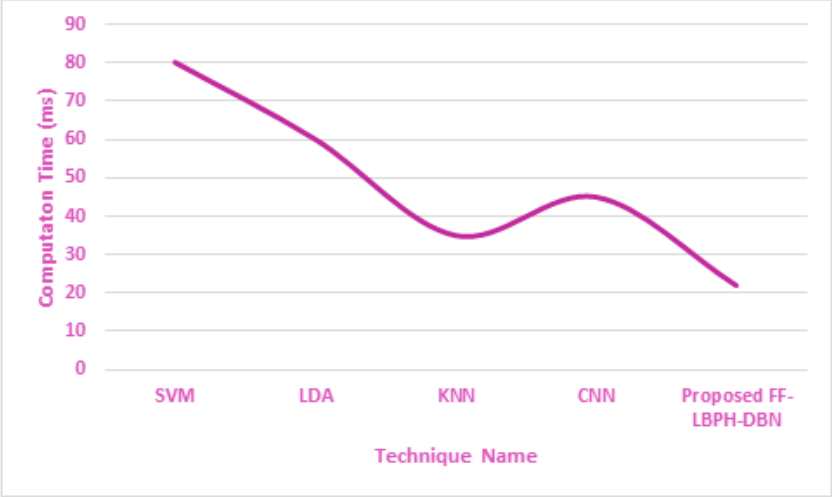

Figure 5 **Computation time.**

Full-size DOI: 10.7717/peerjcs.881/fig-5

CONCLUSION

In this paper, the fisherface Linear binary pattern histogram using the DBN classifier (FF-LBPH DBN) technique was implemented as a detection technique for deepfake images. The proposed work was faster in execution and the detection of fake image and real image was very effective. Deepfake face image manipulations were analyzed using FF-LBPH DBN model and it also produced high level of accuracy. The pre-processing work had been done using Kalman filter for the detection of the fake images in a fine-tuned recognition. In order to get less execution of time, the dimensionality reduction of features were utilized

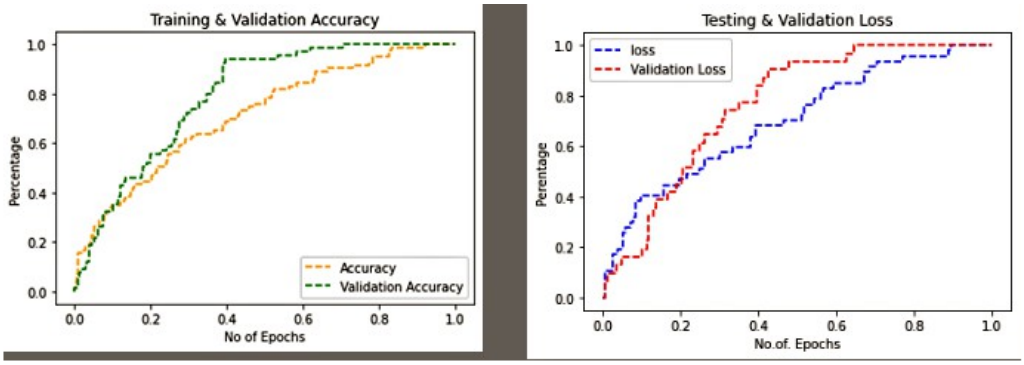

Figure 6 **Training & validation accuracy and loss in proposed work.**
Full-size DOI: 10.7717/peerjcs.881/fig-6

using a fusion of the fisherface-LBPH algorithm. It helped in detecting the fake face image which in turn could prevent the individuals from being defamed unknowingly. From the results, it was concluded that the proposed work FF-LBPH had produced better detection and analysis of deepfake face image. The accuracy rate of proposed work FF-LBPH-DBN had attained a value of 98.82% in the CASIA-WebFace image dataset. The next position in terms of accuracy rate was 97.82% for the DFFD dataset. For future work, it may be extended up to various classifiers and use of different distance metric measures for detecting the deepfake face image.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding
The authors received no funding for this work.

Competing Interests
The authors declare there are no competing interests.

Author Contributions
- Suganthi St conceived and designed the experiments, authored or reviewed drafts of the paper, and approved the final draft.
- Mohamed Uvaze Ahamed Ayoobkhan conceived and designed the experiments, prepared figures and/or tables, and approved the final draft.
- Krishna Kumar V conceived and designed the experiments, analyzed the data, prepared figures and/or tables, and approved the final draft.
- Nebojsa Bacanin performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, and approved the final draft.
- Venkatachalam K performed the experiments, analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.
- Hubálovský Štěpán and Trojovský Pavel performed the experiments, performed the computation work, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The data is available at Kaggle: Available at https://www.kaggle.com/arnaud58/flickrfaceshq-dataset-ffhq.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.881#supplemental-information.

REFERENCES

Afchar D, Nozick V, Yamagishi J, Echizen I. 2018. Mesonet: a compact facial video forgery detection network. In: *2018 IEEE international workshop on information forensics and security (WIFS)*. Piscataway: IEEE, 1–7.

Allcott H, Gentzkow M. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives* **31(2)**:211–36 DOI 10.1257/jep.31.2.211.

Arasaratnam I, Haykin S, Hurd TR. 2010. Cubature Kalman filtering for continuous-discrete systems: theory and simulations. *IEEE Transactions on Signal Processing* **58(10)**:4977–4993 DOI 10.1109/TSP.2010.2056923.

Arunkumar P, Kannimuthu S. 2020. Mining big data streams using business analytics tools: a birdâĂŹs eye view on moa and samoa. *International Journal of Business Intelligence and Data Mining* **17(2)**:226–236.

Badale A, Castelino L, Darekar C, Gomes J. 2018. Deep fake detection using neural networks. In: *15th IEEE international conference on advanced video and signal based surveillance (AVSS)*. Piscataway: IEEE.

Cellan-Jones R. 2019. Deepfake videos double in nine months. *BBC News. Available at https://www.bbc.com/news/technology-49961089#:~:text=New%20research%20shows%20an%20alarming,is%20becoming%20a%20lucrative%20business*.

Chen P, Liu J, Liang T, Zhou G, Gao H, Dai J, Han J. 2020. Fsspotter: spotting face-swapped video by spatial and temporal clues. In: *2020 IEEE international conference on multimedia and expo (ICME)*. Piscataway: IEEE, 1–6.

Citron DK. 2019. How deepfakes undermine truth and threaten democracy. *Available at https://www. Ted.com/talks/danielle_citron_how_deepfakes_undermine_truth_and_threaten_democracy?language=en*.

Dang H, Liu F, Stehouwer J, Liu X, Jain AK. 2020. On the detection of digital face manipulation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. Piscataway: IEEE, 5781–5790.

Guarnera L, Giudice O, Battiato S. 2020. Deepfake detection by analyzing convolutional traces. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. Piscataway: IEEE, 666–667.

Hinton GE, Osindero S, Teh Y-W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* **18(7)**:1527–1554 DOI 10.1162/neco.2006.18.7.1527.

Hulzebosch N, Ibrahimi S, Worring M. 2020. Detecting cnn-generated facial images in real-world scenarios. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. Piscataway: IEEE, 642–643.

Jagdale R, Shah S. 2019. A novel algorithm for video super-resolution. In: *Information and communication technology for intelligent systems*. Singapore: Springer, 533–544.

Kalman RE. 1960. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* **82(1)**:35–45 DOI 10.1115/1.3662552.

Korshunov P, Marcel S. 2018. Deepfakes: a new threat to face recognition? assessment and detection. ArXiv preprint. arXiv:arXiv:1812.08685.

Lazer DM, Baum MA, Benkler Y, Berinsky AJ, Greenhill KM, Menczer F, Metzger MJ, Nyhan B, Pennycook G, Rothschild D , et al. 2018. The science of fake news. *Science* **359(6380)**:1094–1096 DOI 10.1126/science.aao2998.

Li L, Bao J, Zhang T, Yang H, Chen D, Wen F, Guo B. 2020. Face x-ray for more general face forgery detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. Piscataway: IEEE, 5001–5010.

Li Y, Lyu S. 2018. Exposing deepfake videos by detecting face warping artifacts. ArXiv preprint. arXiv:arXiv:1811.00656.

Mahendhiran P, Kannimuthu S. 2018. Deep learning techniques for polarity classification in multimodal sentiment analysis. *International Journal of Information Technology & Decision Making* **17(03)**:883–910.

Maheswaran S, Kuppusamy P, Ramesh S, Sundararajan T, Yupapin P. 2018. Refractive index sensor using dual core photonic crystal fiber–glucose detection applications. *Results in Physics* **11**:577–578 DOI 10.1016/j.rinp.2018.09.055.

Maheswaran S, Sathesh S, Priyadharshini P, Vivek B. 2017. Identification of artificially ripened fruits using smart phones. In: *2017 international conference on intelligent computing and control (I2C2)*. Piscataway: IEEE, 1–6.

Maras M.-H, Alexandrou A. 2019. Determining authenticity of video evidence in the age of artificial intelligence and in the wake of deepfake videos. *The International Journal of Evidence & Proof* **23(3)**:255–262 DOI 10.1177/1365712718807226.

Marra F, Saltori C, Boato G, Verdoliva L. 2019. Incremental learning for the detection and classification of gan-generated images. In: *2019 IEEE international workshop on information forensics and security (WIFS)*. Piscataway: IEEE, 1–6.

Nataraj L, Mohammed TM, Manjunath B, Chandrasekaran S, Flenner A, Bappy JH, Roy-Chowdhury AK. 2019. Detecting GAN generated fake images using co-occurrence matrices. *Electronic Imaging* **2019(5)**:532–1 DOI 10.2352/ISSN.2470-1173.2019.5.MWSF-532.

Neves JC, Tolosana R, Vera-Rodriguez R, Lopes V, Proença H, Fierrez J. 2020. Ganprintr: Improved fakes and evaluation of the state of the art in face manipulation detection. *IEEE Journal of Selected Topics in Signal Processing* **14(5)**:1038–1048 DOI 10.1109/JSTSP.2020.3007250.

Nguyen HM, Derakhshani R. 2020. Eyebrow recognition for identifying deepfake videos. In: *2020 international conference of the biometrics special interest group (BIOSIG)*. Piscataway: IEEE, 1–5.

Ranjan P, Patil S, Kazi F. 2020. Improved generalizability of deep-fakes detection using transfer learning based CNN framework. In: *2020 3rd international conference on information and computer technologies (ICICT)*. Piscataway: IEEE, 86–90.

Tolosana R, Vera-Rodriguez R, Fierrez J, Morales A, Ortega-Garcia J. 2020. Deepfakes and beyond: a survey of face manipulation and fake detection. *Information Fusion* **64**:131–148 DOI 10.1016/j.inffus.2020.06.014.

Vivek B, Maheswaran S, Keerthana P, Sathesh S, Bringeraj S, Sri RA, Sulthana SA. 2018. Low cost raspberry pi oscilloscope. Piscataway: IEEE, 386–390.

Wang R, Ma L, Juefei-Xu F, Xie X, Wang J, Liu Y. 2019. FakeSpotter: a simple baseline for spotting AI-synthesized fake faces. ArXiv preprint. arXiv:1909.06122.

Yadav D, Salmani S. 2019. Deepfake: a survey on facial forgery technique using generative adversarial network. In: *2019 International conference on intelligent computing and control systems (ICCS)*. Piscataway: IEEE, 852–857.

Yang X, Li Y, Lyu S. 2019. Exposing deep fakes using inconsistent head poses. In: *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. Piscataway: IEEE, 8261–8265.

Yu N, Davis L, Fritz M. 2018. Attributing fake images to gans: analyzing fingerprints in generated images. ArXiv preprint. arXiv:arXiv:1811.08180.