



# A hybrid method for biometric authentication-oriented face detection using autoregressive model with Bayes Backpropagation Neural Network

M. Vasanthi<sup>1</sup> · K. Seetharaman<sup>2</sup>

Published online: 13 January 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH, DE part of Springer Nature 2021

## Abstract

This paper proposes a novel method, which is coined as ARBBPNN, for biometric-oriented face detection, based on autoregressive model with Bayes backpropagation neural network (BBPNN). Firstly, the given input colour key face image is modelled to HSV and YCbCr models. A hybrid model, called HS–YCbCr, is formulated based on the HSV and YCbCr models. The submodel,  $H$ , is divided into various sliding windows of size,  $3 \times 3$ . The model parameters are estimated for the window using the BBPNN. Based on the model coefficients, autocorrelation coefficients (ACCs) are computed. An autocorrelation test tests the significance of the ACCs. If the ACC passes the test, then it is inferred that the small image region, viz. the window, represents the texture and it is treated as the texture feature. Otherwise, it is regarded as structure, which is treated as the shape feature. The texture and shape features are formulated as feature vectors (FV) separately, and they are combined into a single FV. This process is performed for all colour submodels. The FVs of the submodels are combined into a single holistic vector, which is treated as the FV of the key face image. The key FV has twenty feature elements. The similarity of the key and target face images is examined, based on the key and target FVs, by deploying multivariate parametric statistical tests. If the FVs of the key and target images pass the tests, then it is concluded that the key and target face images are the same; otherwise, they are regarded as different. The GT, FSW, Pointing'04, and BioID datasets are considered for the experiments. In addition to the above datasets, we have constructed a dataset with face images collected from Google, and many images captured through a digital camera. It is also subjected to the experiment. The obtained recognition results show that the proposed ARBBPNN method outperforms the existing methods.

**Keywords** Face recognition · Shape feature · Texture feature · Autocorrelation coefficient · Multivariate parametric test

## 1 Introduction

With the advent of the advanced digital technologies, such as ICT, IoT, WoT, mass volume of storage devices, and fast computational methods, the biometric system has

tremendously evolved during the last two decades. Nowadays, almost all the organizations, irrespective of size and installation capability, use the biometric-driven security systems. In biometric systems, especially in the public-security-driven biometric system, face recognition plays a vital role in identifying the criminals and preventing the crime and its related activities in public. Also, it has got proper attention of research communities of computer vision and pattern recognition.

Specifically, during and after the global pandemic Covid-19, the facial-based biometric system has skewed the attention of the biometric technology vendors because the users in worldwide are avoiding fingerprint and other contact-based biometric systems. Thus, worldwide organisations that had implemented the contact-based biometric systems have turned on to non-touch facial authentication platform as a result of Covid-19. Hence, in the future, there

---

Communicated by V. Loia.

✉ K. Seetharaman  
kseethaddeau@gmail.com

M. Vasanthi  
wmsami@kku.edu.sa

<sup>1</sup> Department of Computer Science, King Khalid University, Abha, Kingdom of Saudi Arabia

<sup>2</sup> Department of Computer and Information Science, Annamalai University, Annamalai Nagar, Tamil Nadu 608 002, India

will be a spike in demand for non-contact biometric systems in organisations. Therefore, this paper believes that it will reflect in research organisations, which involved in biometric technology-oriented systems development. Moreover, it will induce the biometric research community to concentrate more on contactless biometric systems development like facial, iris, gait, and voice recognition (Carlaw 2020; News 2020a).

In face recognition, the features and feature extraction methods play a significant role. They have been broadly categorized into global and local. The local methods are too effective than the global for face recognition, even if it is a minute feature (Singh and Chabra 2018). Several local feature extraction methods, such as local binary pattern (LBP), local ternary pattern (LTD), local derivative pattern (LDP), Weber's local descriptor (WLB), histogram of oriented gradient (HOG), have been proposed by many researchers (Shi et al. 2010; Tan and Triggs 2010; Chai et al. 2014; Singh and Chabra 2018; He et al. 2018; Muqet and Holambe 2019) for face recognition. They have reported that the local feature extraction methods yield better results than the global methods. In recent years, ordinal measures have drawn the attention of the research community of facial recognition (Chai et al. 2014; He et al. 2018). The binary-code-driven ordinal features are robust to intra-class variations during binarization (He et al. 2018). He et al. suggest, in biometric systems, an ordinal measure represents the average brightness of two adjacent regions or the relative ordering of two colour channels within the same region. The colour feature plays a noteworthy role in facial image representation and recognition. Many researchers have used colour features at the local level for pattern recognition, especially for image retrieval (Theoharatos et al. 2005; Bindemann and Burton 2009; Liu and Liu 2010; Seetharaman 2015; Pavithra and Sharmila 2017) and face recognition (Liu and Liu 2008; Thakur et al. 2011; Uçar 2014; Pujol et al. 2017; Sharifara et al. 2017); they have reported that the colour features show good results. The literature reveals that spatial orientation feature and texture feature significantly contributed to face recognition (Liu and Liu 2010).

## 2 Related works

Liu and Liu (2008) have presented a method based on a combination of colour feature and frequency feature for face identification. They have combined the R component of the RGB colour model, and I and Q components of the YIQ colour model; reported that the proposed hybrid method gives a good result. Thakur et al. (2011) have introduced a skin colour model, RGB-HS-CbCr, for face identification, and they report that the HS, Cb, and Cr

components yield better results than the RGB colour model. Thus, they use a set of bounding rules, based on the colour distribution to detect the boundary of the human face. Terrillon et al. (2000) have performed a comparative study of nine different colour models and reported that the HSV and YCbCr models result in better segmentation than the others. Wang et al. (2018) have adopted the YCbCr model as it has excellent segregation characteristics of the skin tone and background scenes, and they have reported that it demands minimal computational cost than the HSV colour model.

The LBP, LDP, LTD, and HOG encode the image pixels, based on the positional number system. The positional number-based encoding system could lead to wrong encoding and results in less precision. For example, LBP and LTP methods assign the weights 1 and 128 to the pixels at locations (1, 1) and (3, 3) of the sub-image of size  $(3 \times 3)$ , respectively. It is not correct because spatially, the pixels at locations (1, 1) and (3, 3) equally contribute to the centre pixel. The descriptors derived in both methods, based on the positional number system, are not rational because it shows a great discrimination on weights given to the pixels at locations (1, 1) and (3, 3). It could lead to a wrong or partial feature description in the small image region.

In order to overcome this problem, this paper presents an autocorrelation (AC)-based feature extraction method, which is the best representation of the spatial information about the pixels. The proposed method builds a hybrid colour model with a combination of H and S of the HSV model and YCbCr model, called HS-YCbCr, and extracts ACC coefficient (ACC) from the HS-YCbCr model. The H and S represent hue and saturation; Y and CbCr represent luminance and chrominance, respectively; Cb and Cr denote blue and red chrominance, respectively. The ACC is computed from each submodel of the HS-YCbCr model. The computed ACC is subjected to the significance test at a specific level of significance to examine the degree of correlation of the pixels in the small image region, called the window. The literature (Hatamikia et al. 2014) reveals that ACC is the best representation of the spatial information about the pixels. Hence, the proposed method deploys ACC-driven feature descriptor for face recognition.

The literature reveals that many researchers have applied statistical models for face detection (Wang et al. 2009; Mil et al. 2012; Yan et al. 2013; Hatamikia et al. 2014; Wu et al. 2017). Wang et al. (2009) have proposed a Bayesian method, based on Markov random field model, for face recognition. They have used Gabor wavelet features, and the relationships between the Gabor features at different pixel locations, which provides higher-order contextual constraints. Hatamikia et al. (2014) have

investigated the performance of autoregressive (AR) features for the classification of emotional states, based on the AR model, sequential forward feature selection, and k-nearest neighbour classifier using EEG signals during emotional audio-visual inductions. They have reported that the AR features outperform the existing methods. Arashloo (2016) has constructed a deep multilayer architecture and undirected graphical models based on Markov random field. Wu et al. (2017) have introduced a coupled hidden Markov random field model for face clustering, face tracking, and their interactions. Liu et al. (2019) have presented a method based on conditional random regression forests, which detects the real-time smiling face detection. They claim that the proposed method is not sensitive to head poses as the relationship between the image patches and smile intensity is conditionally modelled to head pose. The techniques—regression forest, multiple-label dataset augmentation, and non-informative patch removal—have been deployed to achieve high performance.

Paul et al. (2020) have proposed a method, based on geometrical distance and statistical pattern matching of binary facial components: eyes, nose, and mouth regions. It employs the statistical pattern matching tools—standard deviation of Chi-square values with a probability of white pixels (PWPs), standard deviation of HuMIs with Hu's seven-moment invariants, Abs-DifPWPs, and geometrical distance values (GDVs)—for recognition. GDVs are computed between two similar facial corner points (FCPs), and nine FCPs are extracted from whole binary face and BFCs. Pixel intensity values (PIVs) are determined using L2 norms from greyscale values of the whole face and greyscale values of the face components. In this method, the corner points of the eyes, nose, and mouth regions are considered as the geometrical points; GDVs are computed based on the corner points. The GDV-based similarity could lead to wrong results or less precision because they measure the region of the components; the region is computed using the Otsu's algorithm. One cannot regard the regions, which is segmented by Otsu's algorithm, for geometrical points detection because the Otsu's algorithm approximately segments the object or region of interest from the background scenes. Also, they extract statistical features from the greyscale and binary pattern images that are not sufficient and appropriate for face recognition. Because the facial images fully rely on the skin colour chromaticity. Thus, this paper believes that the features extracted from greyscale and binary pattern could not lead to good results or results in with less recognition rate.

On the other hand, in recent years, deep neural networks (DNN) attracted many researchers. However, there are some drawbacks as reported in Goswami et al. (2019) that the DNN architecture-based methods are mostly a black

box method since it is not easy to mathematically formulate the functions that are learned within its many layers of representation. They have examined the robustness of DNNs for face recognition and reported that the performance of deep learning-based face recognition algorithms significantly suffers in the presence of vulnerabilities and adversaries. Wu et al. (2019) reported that the CNN methods use identical operations to all local regions of each face for feature aggregation and give equal weight to all local features for the face detection process. Also, they suggest that these methods not only regard the local features as suboptimal owing to ignoring region-wise distinctions, but also the overall face representations are semantically inconsistent. Realizing this, many researchers have exploited the drawbacks of deep learning-based algorithms that questions their robustness and singularities.

This paper also believes, as suggested by Goswami et al., most of the DNN architectures do not transparently discuss the methods involved in feature selection, feature matching, and other kinds of facial image analyses. Despite there is no second thought of the accuracy of the results attained by DNN, however, one thing is clear that whatever the techniques deployed for recognizing human faces without mathematical conception, it is a challenging and time-consuming process because the human face recognition significantly contributed to mathematical conception. Also, the literature reveals that the DNN methods require more computation cost as it comprises many layers, and this is another disadvantage of the DNN. Hence, to shun these problems mainly computation cost, the proposed method is designed with a Bayes backpropagation neural network (BBPNN), which is deployed only for feature selection, not for the entire face recognition process. Conventional procedures are used for feature matching and face identification. The experimental results show that the proposed method achieves better results with less computation cost than the state-of-the-art methods.

Although many statistical models have been developed for face recognition, the literature reveals that yet to develop a holistic model because the existing models are complementary to each other. It motivated us to develop a holistic method for face recognition; thus, this paper attempts to build such a comprehensive model with less computation cost. The proposed method addresses the drawbacks mentioned above as follows:

1. The proposed method uses HS–YCbCr colour model instead of greyscale and binary image, and extract features from each colour submodel such as H, S, Y, Cb, and Cr.
2. The proposed AR model with Bayes backpropagation neural network (ARBBPNN) method learns deep shape and deep texture features from each colour submodel.

It is observed from the literature that mainly the facial images rely on colour chromaticity and skin textural patterns, and the ACC is the best representation of the spatial orientation of the pixels and textural patterns.

## 2.1 Outline of the proposed method

The proposed ARBBPNN method performs face recognition process in three stages: first, pre-processes the given input image; secondly, extracts the features using the BBPNN method; finally, it performs searching and matching the target face images.

*Pre-process* At first, the given input key image is denoised; then, it is converted to HSV and YCbCr colour models if it is colour. If the key image is greyscale, then it is taken into account for further analysis as it is. The proposed method builds a hybrid colour model, called HS–YCbCr, with a combination of H and S submodels of the HSV model and the YCbCr colour model. Each colour submodel, such as H, S, Y, Cb, and Cr, is segregated into texture and shape components. For example, the denoised and the texture and shape components are presented in Figs. 6 and 7, respectively.

*Feature extraction* The ACC-based feature descriptor is computed from each colour submodel, based on the AR model coefficients. The coefficients are computed using the model parameters. The BBPNN model is employed to estimate the parameters of the AR model. The hidden layer-4 of the BBPNN computes the model coefficients, and the ACCs are calculated using the model coefficients. The hidden layer-5 tests the significance of the ACC, based on the AC test. If the degree of AC is high, then it is inferred that the small image region contributes to texture; otherwise, it is regarded as shape or structure. The ACC with high degree is formulated into a feature vector (FV), called texture feature vector ( $FV^t$ ), and the ACC with low degree is formulated as a separate FV, which is termed as a shape feature vector ( $FV^s$ ). Each FV consists of ten feature elements. The shape-driven and the texture-driven features are merged into a single FV, as shown in Eq. (36), which has twenty FV elements and denoted by  $FV^k$ .

*Searching and matching* The target face image identification is performed in two stages: (1) tests the similarity of the FVs of the key and target face images in terms of variation; (2) tests the distance between the mean vectors of the FVs of the key and target images, provided the images passed the test conducted at stage (1). In order to perform the above two tests, a hypothesis is formulated, i.e. null hypothesis ( $H_0$ ): the key and target face images are the same; otherwise, the alternative hypothesis ( $H_a$ ): the key and target face images differ. At the first stage, the test for equality of covariance of the FV of the key and target

images is performed; at stage two, the test for equality of mean values of the FVs of the key and target images is conducted. If  $H_0$  is accepted, then it is inferred that the key and target face images are the same; otherwise, they differ. The overall functions of the proposed method are illustrated in Fig. 1.

The rest of the paper is organized as follows. Section 3 discusses the pre-processing technique. Section 4 describes the proposed feature selection methods based on the BBPNN method and AR model, while Sect. 5 discusses the test statistics for face recognition. In Sect. 6, the experimental setups and the performance evaluation methods are briefed; the experimental results are demonstrated with suitable diagrams. Section 7 deals with computation cost. Section 8 concludes the papers with a conclusion and provides a future direction for the proposed method.

## 3 Pre-process

There are many possibilities of incorporating different kinds of noises in the images while capturing or manipulating for further analysis. So, the proposed method employs a pre-processing technique, called soft-SVDD (Liu et al. 2014), to remove or mitigate the effects of the noise (abnormality) in the image. The main objective of the SVDD is to identify the noised pixels in the feature space; feature space means the transformed domain of the image. The identified noised pixels are replaced by median values of the normal (except abnormal) pixels of the image. The denoising technique is expressed in Eq. (1), which enriches the proposed face recognition method even if the image is profoundly affected by different noises. In order to identify the abnormal pixels in the image, the given input image is divided into sliding windows of size,  $3 \times 3$ .

$$\|f_{k,l} - C_{k,l}\|^2 \leq r^2 \quad (1)$$

where  $\|\cdot\|$  is the Euclidean norm;  $C_{k,l}$  is the mean intensity value of the pixels in the window, which is computed using the expression in Eq. (2).

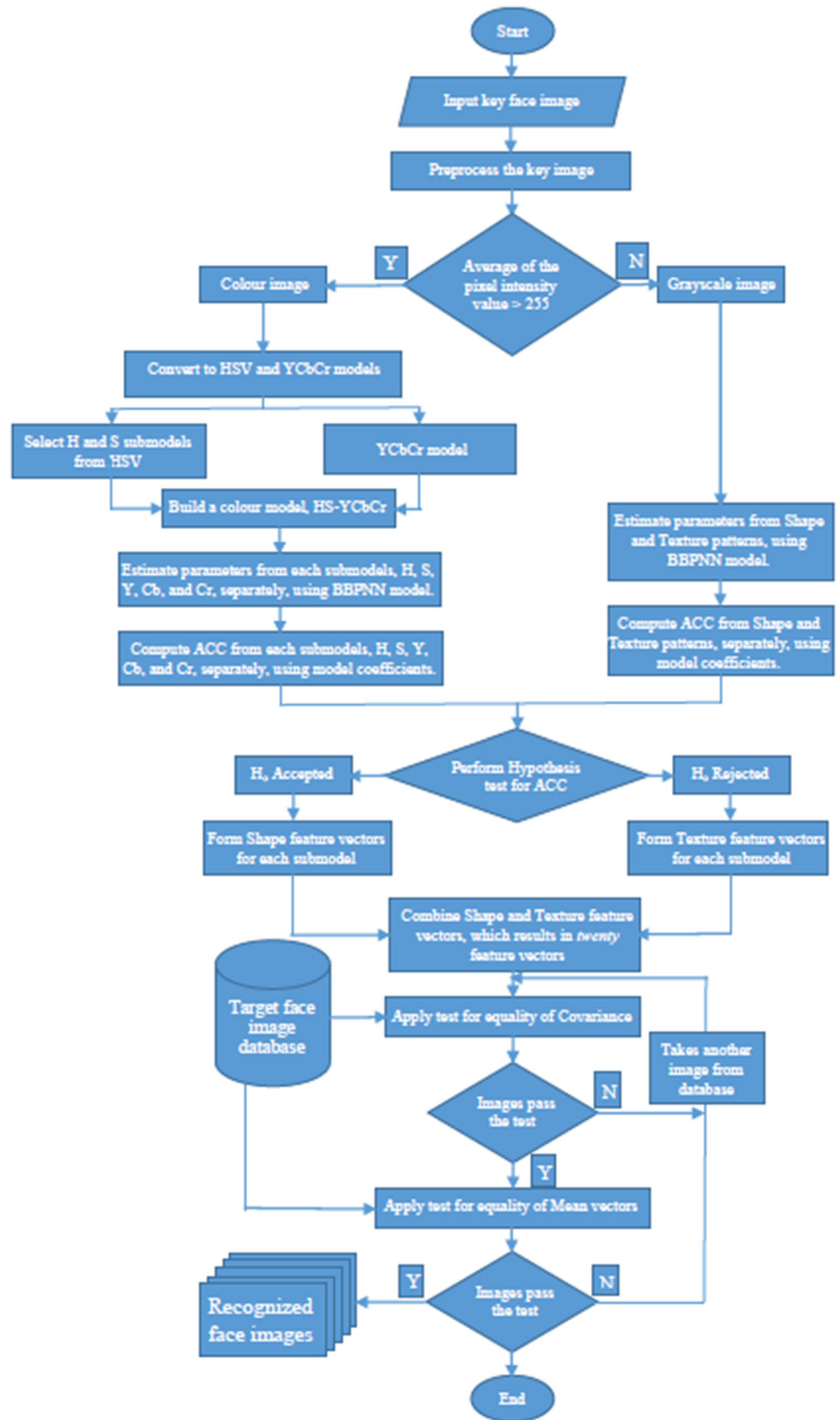
$$C_{k,l} = \sum_{u=1}^{p+n} \lambda_u \theta(f_{k,l}) \quad (2)$$

$\lambda_u$  denotes the Lagrange multiplier;  $\theta(f_{k,l})$  is a vector form of the pixel intensity values;  $f_{k,l}$  is the intensity value of the pixel to be classified, whether it is noised or not.

To classify the pixel,  $f_{k,l}$ , the distance between the  $f_{k,l}$  and  $C_{k,l}$  is computed. If it is less than or equal to radius,  $r$ , that is, it satisfies the condition in Eq. (1), then  $f_{k,l}$  is accepted as a normal pixel; otherwise, it is identified as abnormal.

In order to derive the function in Eq. (1), let  $\{f_{k,l} \in R^n; i, k = 1, 2, 3\}$  be a set of targeted pixels in the

**Fig. 1** Overall functions of the proposed method



window. In which, the functions  $m^{\text{nor}}(f_{k,l}^{\text{nor}})$  and  $m^{\text{neg}}(f_{k,l}^{\text{neg}})$  represent the degree of the membership of a pixel,  $f_{k,l}$ , towards normal class and negative class, respectively. The functions  $f_{k,l}^{\text{nor}}$  and  $f_{k,l}^{\text{neg}}$  contain the group of normal and abnormal pixels, respectively. The solution to the soft-SVDD can be achieved by solving the following optimization problem.

$$\begin{aligned} \min G &= r^2 + \omega_1 \sum m^{\text{nor}}(f_{k,l})\xi_i + \omega_2 m^{\text{neg}}(g_{k,l})\xi_j \\ \text{s. t. } &\|f_{k,l} - C_{k,l}\| \leq r^2 + \xi_i, f_{k,l} \in S_{\text{nor}} \\ &\|g_{k,l} - C_{k,l}\| \leq r^2 + \xi_j, g_{k,l} \in S_{\text{neg}} \\ &\xi_i \geq 0, \xi_j \geq 0, \end{aligned} \tag{3}$$

where  $\omega_1$  and  $\omega_2$  control the trade-off between the pixels in the window and the error;  $\xi_i$  and  $\xi_j$  are defined as a measure of error. The terms  $m^{\text{nor}}(f_{k,l}^{\text{nor}})\xi_i$  and  $m^{\text{neg}}(f_{k,l}^{\text{neg}})\xi_j$  can be regarded as a measure of error with different weighing factors. It is noted that a smaller value of  $m^{\text{nor}}(f_{k,l}^{\text{nor}})$  could reduce the effect of the parameter  $\xi_i$  in Eq. (3), such that the corresponding pixel,  $f_{k,l}^{\text{nor}}$ , becomes less significant.

The decision boundary of the normal and abnormal pixels is constructed by deploying the conditions of the following Lagrange-multiplier’s theorem and the Karush–Kuhn–Tucker complementarity (Vapnik 1998), and then, one can obtain the radius,  $r$ , of the hyper-plane.

**Theorem** *The solution to the problem expressed in Eq. (3) can be arrived by optimizing the problem in Eq. (4):*

$$\begin{aligned} \max &\sum_{i=1}^{m+n} \lambda_i K(m^{\text{nor}}(f_{k,l}^{\text{nor}}), m^{\text{nor}}(f_{k,l}^{\text{nor}})) \\ &- \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} \lambda_i \lambda_j K(m^{\text{nor}}(f_{k,l}^{\text{nor}}), m^{\text{neg}}(f_{k,l}^{\text{neg}})) \\ \text{s. t. } &0 \leq \lambda_i \leq \omega_i^p; \quad i = 1, 2, \dots, m+n, \\ &\sum_{i=1}^{m+n} \lambda_i = 1 \end{aligned} \tag{4}$$

where  $\lambda_i, \lambda_i \geq 0$  are the Lagrange multipliers;  $\{\omega_i^p = \omega_1 m^{\text{nor}}(f_{k,l}^{\text{nor}}), (i = 1, 2, \dots, l)\}$  and  $\{\omega_i^p = \omega_2 m^{\text{neg}}(f_{k,l}^{\text{neg}}), (i = l+1, l+2, \dots, l+n)\}$ .

Let us assume  $f_{k,l}^u$  is one of the patterns of the pixels in the window, and  $r$  can be measured as follows:

$$\begin{aligned} r^2 &= \|f_{k,l}^u - C_{k,l}\|^2 \\ &= (f_{k,l}^u - C_{k,l}) \cdot (f_{k,l}^u - C_{k,l}) \\ r^2 &= (f_{k,l}^u)^2 + (C_{k,l})^2 - 2(f_{k,l}^u) \cdot (C_{k,l}) \end{aligned} \tag{5}$$

Based on the Mercer’s Kernel function, Eq. (5) can be written as in Eq. (6):

$$r^2 = K(f_{k,l}^u, f_{k,l}^u) + K(C_{k,l}, C_{k,l}) - 2K(f_{k,l}^u, C_{k,l}) \tag{6}$$

By applying the Karush–Kuhn–Tucker (Vapnik 1998), one can derive the distance,  $r$ , from the decision boundary of the pixels in the window to its mean intensity values of the pixels.

$$\begin{aligned} r^2 &= K(f_{k,l}^u, f_{k,l}^u) + \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} L_i L_j K(f_{k,l}^{\text{nor}}, f_{k,l}^{\text{neg}}) \\ &- K \sum_{i=1}^{m+n} L_i (f_{k,l}^{\text{neg}}, f_{k,l}^u) \end{aligned} \tag{7}$$

To identify whether a pixel in the window is noised or not, it is enough to compute the distance from the decision boundary of the pixels in the window to the mean value of the pixels in the window. If the distance is less than or equal to  $r$ , that is,

$$\|f_{k,l} - C_{k,l}\|^2 \leq r^2, \tag{8}$$

then it is inferred that the pixel,  $f_{k,l}$ , is normal. Otherwise, it is treated as an abnormal.

For more details about Lagrange-multiplier and Karush–Kuhn–Tucker complementarity, readers can refer to Liu et al. (2014) and Vapnik (1998).

In the sequel, the denoised key face image was converted to HS–YCbCr colour model. The ACC was computed from each colour submodel—H, S, Y, Cb, Cr, based on the sliding window of size  $3 \times 3$ . The computed ACC was tested, whether it is highly correlated or not, using the significance test discussed in Sect. 4.2.1. If the ACC is highly correlated, then it is taken into account of texture features; otherwise, it treated as shape features. For example, the structure and shape of an image are shown in Fig. 7.

### 4 Proposed feature descriptor

Guo et al. (2010) have reported that despite the LBP performs well for pattern recognition; still, it requires more investigation for better results. They have presented a complete model for local binary pattern operator (CLBP) and employed for classification of texture, and claimed that it addresses the above problems. But, still, it needs further investigation because the CLBP method again assigns the

binary values based on the difference between the centre and the neighbouring pixels. Seetharamana and Palanivel (2013) suggested that the assignment of binary and ternary codes, based on the difference between the centre and the adjacent pixels, is not rational. This paper believes that again the problem arises in CLBP as in the case of LBP. In order to overcome this problem, this paper computes AC; the AC coefficient is tested whether it is significantly correlated or not. The computation of the ACC and the significance test are detailed in the proceeding sections.

### 4.1 Spatial-driven descriptor

Let the pixel intensity,  $f$ , at the location  $(k, l)$ , of an image, be a random process at equal interval of discrete-time space,  $t$ . The pixel at location  $(k, l)$  can be denoted by  $f_{k,l}$ . Let us assume that the pixel,  $f_{k,l}$ , be a value generated by a given run of the process at the time,  $t$ . Suppose the process has mean,  $\mu$ , and variance,  $\sigma^2$ , at discrete-time space,  $t$ , for each  $t$ . In the context of image analysis, the dataset is considered as a two-dimensional (2-D) data as depicted in Fig. 2. In this paper, the full-range autoregressive model proposed in Seetharaman (2012) is adopted and reconstructed into an AR model with order two, i.e. AR(2) as in Eq. (9), which is employed to compute the ACC for a small image region of size,  $3 \times 3$ , as depicted in Fig. 2.

$$f_{k,l} = \sum_{p=-1}^1 \sum_{q=-1}^1 \lambda_{\rho} f_{k+p,l+q} + \varepsilon_{p,q}; \tag{9}$$

s. t.  $\rho = |p + q|; p = q \neq 0; k, l = 0$ .

$$\lambda_{\rho} = \frac{K(\sin(\rho\theta) \cos(\rho\phi))}{\alpha^{\rho}}, \quad \text{s. t., } \rho = 1, 2. \tag{10}$$

where  $k, l$  represent the row and column index of the centre pixel, which is treated as origin of the region, of the small image region.

The model coefficients  $\lambda_{\rho}$  are computed as discussed below. According to the following stationarity theorem, the following conditions are imposed on the AR model coefficients as follows.

$$\text{Stationarity} \Leftrightarrow \lambda_1 + \lambda_2 < 1, \quad -\lambda_1 + \lambda_2 < 1, \quad |\lambda_1| < 2, \quad \text{and} \quad |\lambda_2| < 1. \tag{11}$$

*Stationarity theorem* A necessary and sufficient condition for the AR( $p$ ) model to be stationary is that all of the roots of the polynomial in Eq. (12) lie outside the unit circle.

**Fig. 2** Spatial relationships of the pixels

$f_{(-1,-1)}$	$f_{(-1,0)}$	$f_{(-1,1)}$
$f_{(0,-1)}$	$f_{(0,0)}$	$f_{(0,1)}$
$f_{(1,-1)}$	$f_{(1,0)}$	$f_{(1,1)}$

$$\lambda(\beta) = \prod_{i=1}^p (1 - G_i\beta) = 0. \tag{12}$$

The ACC is computed, based on the model coefficients. The computation of the model coefficients is discussed in the next section.

#### 4.1.1 Parameters learning

The proposed BBPNN model is effectively used to estimate the parameters of the AR(2) model. It consists of 6 layers, such as one input and one output layers, and four hidden layers. Its structure is illustrated in Fig. 3. The parameters are learned as follows.

According to Bayes rules, the point estimates of the parameters,  $\alpha, \theta, \phi$ , and  $K$ , are regarded as means of the respective marginal posterior distribution, i.e. posterior means. In a view to minimizing the computational cost, first, the posterior mean of  $\alpha$  is computed. Then,  $\alpha$  is fixed at its posterior mean, and the conditional means of  $\theta$  and  $\phi$  are evaluated. The  $\hat{\alpha}, \hat{\theta}$ , and  $\hat{\phi}$  are set at their posterior means, and then, the conditional mean of  $K$  is estimated. Thus, the estimates are

$$\begin{aligned} \hat{\alpha} &= E(\alpha) \\ \left( \hat{\theta}, \hat{\phi} \right) &= E\left( \theta, \phi \mid \hat{\alpha} \right) \\ \hat{K} &= E\left( K \mid \hat{\alpha}, \hat{\theta}, \hat{\phi} \right) \end{aligned} \tag{13}$$

First, the image is divided into various sliding windows  $w(k, l)$  of size  $3 \times 3$ , where  $k$  ranges from  $-1$  to  $1$ , and estimate the parameters as follows.

*Estimation of  $\alpha$*  First, the hidden layer-1 estimates the posterior mean, based on the seed values received from input layer that are treated as prior information, as follows. It is not a difficult task to set the seed values to the parameters because their ranges are known.

$$\hat{\alpha}_{ij} = E(\alpha_{ij}) = \frac{\sum_i \alpha \times f(i)}{\sum_i f(i)} \tag{14}$$

where  $f(i) = \sum_r \sum_s T(r, s) = e^{-\beta(\alpha-1)} C^{-d} \alpha^{-1/2}$

$$d = \frac{TP}{2}; \quad \Delta = \frac{\pi}{TP}; \quad \alpha = \frac{\beta}{2}; \quad \beta = \frac{\text{Var}(f_i)}{\bar{f}}$$

where  $T(r, s)$  is a matrix with order  $p \times q$ ; TP is the number of pixels in the window, i.e. TP = 9;  $\alpha$  is a weight factor;  $f(i)$  is the likelihood of observations;  $\sum_i f(i)$  is a normalizing factor.

The  $\lambda_{\rho}$ s ( $\rho = 1, 2, \dots$ ) are computed by substituting  $\hat{\alpha}_{ij}$  and  $\theta, \phi, K$  in Eq. (10), for example, which is exemplified

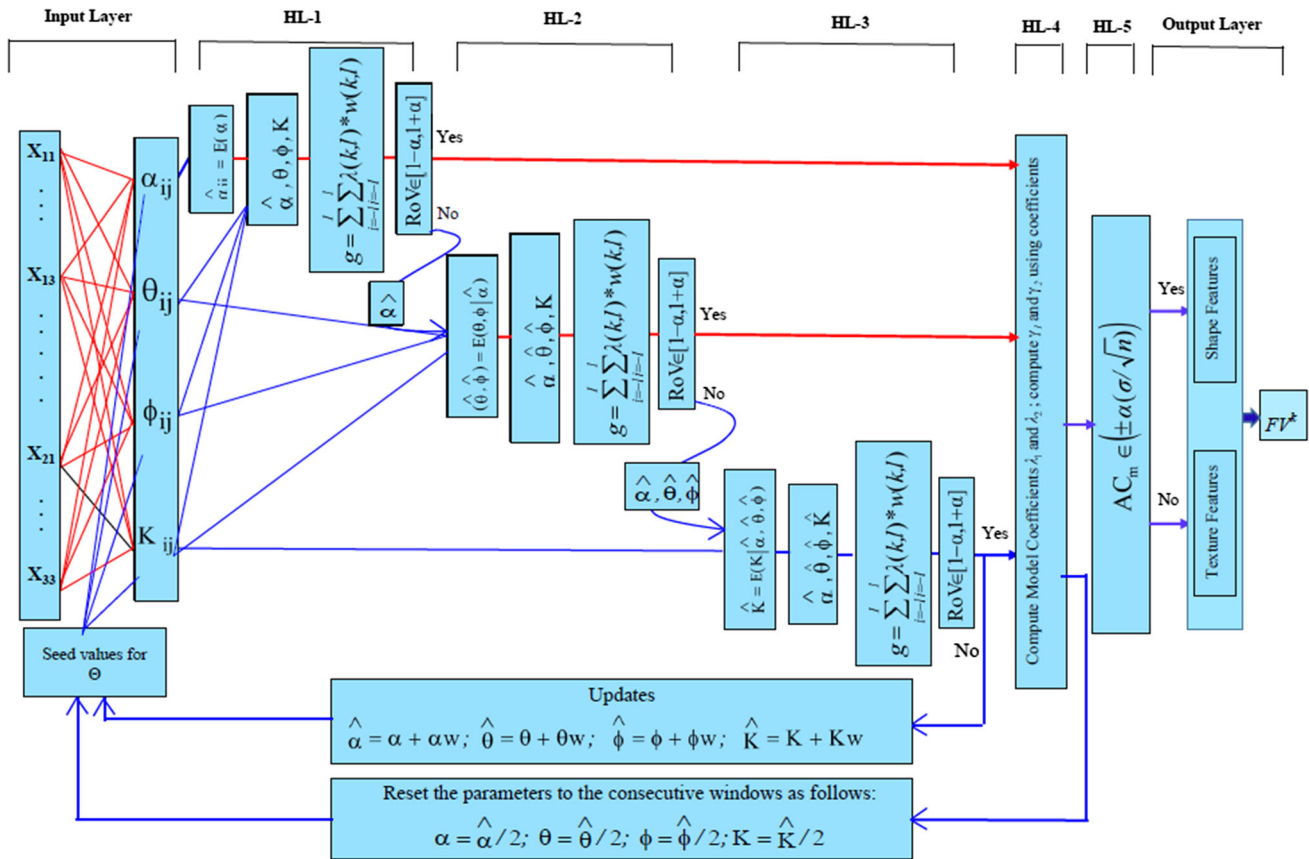


Fig. 3 Structure of the BBPNN with three-layer perceptron

in Eq. (18);  $\hat{\alpha}$  is estimated in hidden layer 1, and the parameters,  $\theta, \phi, K$ , are received from input layer. The computed  $\lambda_p$ s are stored in  $\lambda(k, l)$  as illustrated in Eq. (15). In order to learn  $\lambda_p, \lambda(k, l)$  and  $w(k, l)$  are convolved as depicted in Eq. (16). The  $R$  is computed using Eq. (17). If  $R$  falls in the interval mentioned in Eq. (17), then the iteration process is terminated, and the estimated parameters are sent to hidden layer 4. Otherwise,  $\hat{\alpha}$  is sent to the hidden layer 2.

$$\lambda(k, l) = \begin{bmatrix} \lambda_2 & \lambda_1 & \lambda_2 \\ \lambda_1 & & \lambda_1 \\ \lambda_2 & \lambda_1 & \lambda_2 \end{bmatrix} \quad (15)$$

$$g = \sum_{i=-1}^1 \sum_{j=-1}^1 \lambda(k, l) * w(k, l); \quad \text{s. t. } k = l \neq 0 \quad (16)$$

$$R = \frac{w(k, l)}{g} \in [0.95, 1.05] \quad (17)$$

$$\lambda_1 = \frac{K(\sin(1\theta) \cos(1\phi))}{\alpha^1} \quad \text{and} \quad \lambda_2 = \frac{K(\sin(2\theta) \cos(2\phi))}{\alpha^2} \quad (18)$$

while  $R$  falls in the interval given in Eq. (17), the iteration process is saturated; it means that the predicted pixel has become the same or nearer to the actual pixel.

*Estimation of  $\theta$  and  $\phi$  (hidden layer 2)* The  $\alpha$  is fixed at its posterior mean,  $\hat{\alpha}$ , as in Eq. (13), then  $\theta$  and  $\phi$  are learned, based on the expressions in Eq. (19), using the seed values of  $\theta$  and  $\phi$  received from the input layer that are regarded as prior information.

$$\hat{\theta} = E(\theta) = \frac{\sum_i \theta \times f(i)}{\sum_i f(i)}; \quad \hat{\phi} = E(\phi) = \frac{\sum_i \phi \times f(i)}{\sum_i f(i)} \quad (19)$$

The hidden layer-2 computes  $\lambda_p$ , based on  $\hat{\alpha}, \hat{\theta}, \hat{\phi}, K$ , and stored as in Eq. (15); this layer learns the parameters,  $\hat{\theta}$  and  $\hat{\phi}$ , while  $\hat{\alpha}$  and  $K$  are received from the hidden layer-1 and input layer, respectively. The value of  $g$  is computed as discussed above and applies in Eq. (16); examines whether  $R$  falls in the interval  $[0.95, 1.05]$ . If  $R \in [0.95, 1.05]$ , then the iteration process is terminated, and the estimated parameters are sent to hidden layer 4. Otherwise,  $\hat{\alpha}, \hat{\theta}$ , and  $\hat{\phi}$  are fed as input to hidden layer-3.



*Estimate of K* (hidden layer 3) The  $\alpha$ ,  $\theta$ , and  $\phi$  are fixed at their respective posterior means,  $\hat{\alpha}$ ,  $\hat{\theta}$ , and  $\hat{\phi}$ ; then,  $K$  is learned, based on the expression in Eq. (20), using the seed value of  $K$  received from the input layer that is regarded as prior information.

$$\hat{K} = b/a \tag{20}$$

Layer-3 computes  $R$ , based on  $\hat{\alpha}$ ,  $\hat{\theta}$ ,  $\hat{\phi}$ ,  $\hat{K}$ , and examines whether  $R$  falls in the interval  $[0.95, 1.05]$ . If  $R$  falls in the interval, then the iteration process is terminated, and the estimated parameters are sent to hidden layer 4. Otherwise, the sliding window moves to  $(k + 1)$ th position with half of the values of the parameters estimated in  $k$ th position; and the weight factor is updated for each parameter as in Eq. (21) until the function in Eq. (17) satisfies the condition. The above process is repeated for the entire image. Ultimately, the BBPNN derives a tensor matrix  $F(I, J, K)$  with ACC values, which is output of the BBPNN model.

$$\left. \begin{aligned} \hat{\alpha}_i &= \hat{\alpha}_{i-1} + \Delta_i \\ \hat{\theta}_i &= \hat{\theta}_{i-1} + \Delta_i \\ \hat{\phi}_i &= \hat{\phi}_{i-1} + \Delta_i \\ \hat{K}_i &= \hat{K}_{i-1} + \Delta_i \end{aligned} \right\} \tag{21}$$

where  $i$  denotes the  $i$ th iteration of updating the parameters;  $\Delta_i$  is a very small quantity.

*Hidden layer 4* Based on the estimated parameters, it computes the model coefficients  $\lambda_1$  and  $\lambda_2$ , as exemplified in Eq. (18). The ACCs,  $\rho_1$ , and  $\rho_2$ , are computed by substituting the model coefficients in Eqs. (29) and (30), and the ACCs are sent to the hidden layer-5.

*Hidden layer 5* This layer performs a significance test on computed ACCs, based on the test statistic expressed in Eq. (31). It computes the test statistics,  $AC_m$ , based on the computed AC and tests whether it falls within the confidence limit,  $\pm\alpha(\sigma/\sqrt{n})$ . If the  $AC_m$  falls within the limit, then it is inferred that the ACCs are not highly significant, viz. the small image region of the window contains shape feature, and it is formulated as a FV as depicted in Eq. (34); otherwise, it is assumed that it contains texture feature and formulated as a FV as demonstrated in Eq. (35). The formulated FVs are sent to the output layer. The output layer outputs the FVs.

### 4.2 Basis for computation of ACC

In order to compute ACC, we have to compute the autocovariance for the AR(2) process. More general form of the autocovariance function can be written as in Eq. (22):

$$\varpi_{|p+q|} = E(f_k.f_{k+p,l+q}) \tag{22}$$

where  $p$  and  $q$  are the lag variables in row and column directions of the 2D image pixel intensity values, respectively. In the case of the proposed study, Eq. (22) can be written as follows:

$$\begin{aligned} \varpi_\rho &= E\left\{ \sum_p \sum_q \lambda_\rho (f_k.f_{k-p,j+q}) \right\} + E(\varepsilon_{k,l}\varepsilon_{k+p,l+q}); \\ \text{s. t. } p &= -1, 0, 1; \quad q = -1, 0, 1. \end{aligned} \tag{23}$$

$$\begin{aligned} \varpi_\rho &= E\{ \lambda_\rho [f_{0,0}(f_{-1,0} + f_{0,1} + f_{1,0} + f_{0,-1})] \} \\ &\quad + E\{ \lambda_\rho [f_{0,0}(f_{-1,-1} + f_{-1,1} + f_{1,1} + f_{1,-1})] \} + E(\varepsilon_{0,0}\varepsilon_{p,q}) \\ \text{s. t. } p &= -1 \text{ to } 1; \quad q = -1 \text{ to } 1 \end{aligned} \tag{24}$$

$$\varpi_r = \lambda_1 \varpi_{r-1} + \lambda_2 \varpi_{r-2} \tag{25}$$

where  $\rho = |p + q|$ ,  $\rho$  represents order of the autocovariance function.

The AC function expressed in Eq. (26) can be obtained by dividing Eq. (24) by the variation between the pixel values of the subimage,  $f_{k,l}$ .

$\text{Var}(f_{i,j}) = \frac{\sum_{i=1}^3 \sum_{j=1}^3 f_{i,j}}{N-1}$ , where  $N$  is a number of pixels in the small image region of size,  $3 \times 3$ .

$$\rho_r = \lambda_1 \rho_{r-1} + \lambda_2 \rho_{r-2} \tag{26}$$

Since  $\rho_1 = \rho_{-1}$ , Eq. (26) can be written as:

$$\rho_1 = \lambda_1 + \lambda_2 \rho_{-1} \tag{27}$$

$$\rho_1 = \lambda_1 + \lambda_2 \rho_1 \tag{28}$$

$$\rho_1 = \frac{\lambda_1}{1 - \lambda_2} \tag{29}$$

Similarly, the second-order ACC can be computed as follows:

$$\rho_2 = \frac{\lambda_1^2 + \lambda_1 + \lambda_2^2}{1 - \lambda_2} \tag{30}$$

where  $\rho_1$  and  $\rho_2$  represent the first- and second-order ACC, respectively.

#### 4.2.1 Significance test for ACC

In order to test the significance of the ACC in a small image region of size  $3 \times 3$ , the proposed method adopts the test statistic expressed in Eq. (31), which is presented by Pena and Rodriguez (2002).

$$AC_m = \left[ 1 - |\tilde{\rho}_m|^{\frac{1}{m}} \right] \tag{31}$$

where  $\tilde{\rho}_m$  is the correlation matrix, which is formulated using the standardized ACC,  $\tilde{\rho}_m$ ,  $m$  is the order of the AC. The correlation matrix is expressed in Eq. (32):

$$\tilde{\rho}_m = \begin{bmatrix} 1 & \tilde{r}_1 & \dots & \tilde{r}_m \\ \tilde{r}_1 & 1 & \dots & \tilde{r}_{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{r}_m & \tilde{r}_{m-1} & \dots & 1 \end{bmatrix}; \quad \tilde{r}_m^2 = \frac{(n+2)}{(n+k)} r_m^2 \tag{32}$$

where  $n$  is the number of samples and  $m$  is the lag variable.

The significance test statistic,  $AC_m$ , is performed based on the confidence measure,  $\pm\alpha(\sigma/\sqrt{n})$ , where  $\alpha$  represents significance level;  $\sigma$  denotes standard deviation. To perform the confidence test at a given significance level  $\alpha$ , under  $H_0$  that there is no AC at lag  $m$ , the computed  $AC_m$  is compared to  $(-\alpha(\sigma/\sqrt{n}), +\alpha(\sigma/\sqrt{n}))$ . If the  $AC_m$  falls outside the interval, then  $H_0$  is rejected at the level of significance  $\alpha$ , i.e. the image pixels are correlated; otherwise, the  $H_0$  is accepted, i.e. the image pixels are not correlated. If the AC is high, then it is assumed that the small image region represents texture feature; otherwise, the region represents shape feature. If the AC test is accepted at significance level,  $\alpha$ , then the ACC value is stored as a vector as depicted in Eq. (33), which is treated as shape features; otherwise, it is treated as a texture feature and stored as a vector as depicted in Eq. (34).

The above process is repeated for all submodels (H, S, Y, Cb, and Cr), and finally, the  $FV^{ks}$  and  $FV^{kt}$  are merged into a single holistic FV of the key face image, which is denoted as  $FV^k$ .

$$\overrightarrow{FV}^{ks} = \left\{ \overline{H}^s(\rho_{1i}, \rho_{2i}), \overline{S}^s(\rho_{1i}, \rho_{2i}), \overline{Y}^s(\rho_{1i}, \rho_{2i}), \overline{Cb}^s(\rho_{1i}, \rho_{2i}), \overline{Cr}^s(\rho_{1i}, \rho_{2i}) \right\} \tag{33}$$

$$\overrightarrow{FV}^{kt} = \left\{ \overline{H}^t(\rho_{1i}, \rho_{2i}), \overline{S}^t(\rho_{1i}, \rho_{2i}), \overline{Y}^t(\rho_{1i}, \rho_{2i}), \overline{Cb}^t(\rho_{1i}, \rho_{2i}), \overline{Cr}^t(\rho_{1i}, \rho_{2i}) \right\} \tag{34}$$

where  $FV^{ks}$  and  $FV^{kt}$  denote shape and texture FVs of the key face image, respectively;  $i$  represents the number of sliding windows (number of samples of the features) of the face image.

Similarly, the FVs for target face images of the face image database are computed, which is denoted by  $FV^{tr}$ , and stored in the FV database of the face image. The formulation of the  $FV^{tr}$  is discussed in Sect. 6.

$$\overrightarrow{FV}^k = (\rho_{1i}^{Hs}, \rho_{2i}^{Hs}, \rho_{1i}^{Ss}, \rho_{2i}^{Ss}, \rho_{1i}^{Ys}, \rho_{2i}^{Ys}, \rho_{1i}^{Cbs}, \rho_{2i}^{Cbs}, \rho_{1i}^{Crs}, \rho_{2i}^{Crs}, \rho_{1i}^{Ht}, \rho_{2i}^{Ht}, \rho_{1i}^{St}, \rho_{2i}^{St}, \rho_{1i}^{Yt}, \rho_{2i}^{Yt}, \rho_{1i}^{Cbt}, \rho_{2i}^{Cbt}, \rho_{1i}^{Crt}, \rho_{2i}^{Crt}) \tag{35}$$

### 5 Proposed face recognition test statistics

The combined  $FV^k$  in Eq. (35) is treated as a feature space, which is measurable. Let us assume that the FV space is a sigma field ( $\Omega$ ). The shape and texture features are regarded as a subfield of  $\Omega$ . The shape and texture subspaces are denoted by  $S$  and  $T$ , respectively, such that  $S \subset \Omega$  and  $T \subset \Omega$ . Also, it satisfies  $S \cup T \in \Omega$  and  $S \cap T \in \Omega$ . The subspaces,  $S$  and  $T$ , are assumed as enfolded Gaussian random fields (Carbó-Dorca and Besalú 2011; Kukush 2019). Let  $F^{ki}$  and  $F^{tri}$  ( $i = 1, 2, \dots, 20$ ) be the multidimensional Gaussian distribution functions that can be denoted as  $F^{ki} \sim N(M_k, W_k)$  and  $F^{tri} \sim N(M_t, W_t)$  with mean vector  $M_{(\cdot)}$  and covariance matrix  $W_{(\cdot)}$ . Each feature space, such as shape and texture, has five submodels that are extracted from HS–YCbCr colour model; each submodel has two FVs, such as  $\rho_1$  and  $\rho_2$ , as illustrated in Eq. (35). As a result, the FV consists of twenty FV elements, which is expressed in Eq. (35).

In order to identify the target face image, the multivariate parametric tests, such as the test for equality of covariance matrices and the test for equality of mean vectors, are employed. The test for equality of covariance between the  $FV^k$  and  $FV^{tr}$  is expressed in Eq. (36).

Let  $f_x^{(g)}$ ;  $\alpha = k, tr$ ;  $g = 1, 2, \dots, 10$  be an observation from  $g$ th FV of the key ( $k$ ) face image or the target ( $tr$ ) image which follows Gaussian distribution with mean vector  $M_{(\cdot)}$  and covariance matrix  $W_{(\cdot)}$ . According to the law of distribution that can be denoted as  $F_{ki} \sim N(M_k, W_k)$  and  $F_{tr} \sim N(M_{tr}, W_{tr})$ .

#### 5.1 Test for equality of variation between key and target face images

To test whether the covariance matrices of the key and target images are the same or not, the hypothesis is formulated as follows:

- $H_0$  :  $\Sigma_k = \Sigma_{tr}$  Null hypothesis
- $H_a$  :  $\Sigma_k \neq \Sigma_{tr}$  Alternative hypothesis

The equality of covariance matrices of the key and target face images with 20-feature components of multivariate normal populations can be examined against the alternative of general positive definite matrices. The test statistic given in Eq. (36) interprets the similarity (distance) between the key and target face images, which approximately distributed to Chi-square distribution with degrees of freedom  $1/2(q - 1)p(p + 1)$  (Anderson 2003).

$$D_{Cov} = \left( \sum n_g \ln|W| - \sum_{g=1}^q n_g \ln W_g \right) C^{-1} \tag{36}$$

where

$$\begin{aligned} W_g &= \frac{1}{N_g} S_g, \\ S_g &= \sum_{\alpha} \left( f_{\alpha}^g - \bar{f}^{(g)} \right)^T \left( f_{\alpha}^g - \bar{f}^{(g)} \right); \quad \alpha = k, tr \\ \bar{f}^{(g)} &= \frac{1}{N_{\alpha}} \sum_{\alpha} f_{\alpha}^{(g)}; \quad \alpha = k, tr \\ C^{-1} &= 1 - \frac{2g^2 + 3g - 1}{6(g + 1)(\alpha - 1)} \left( \sum_{\alpha=1}^2 \frac{1}{n_{\alpha}} - \frac{1}{\sum n_{\alpha}} \right) \end{aligned} \tag{37}$$

where  $S_g$  is a sum of squares of cross-products of variation about the mean of the components in a sample, which is drawn from the multivariate normal population;  $N_g$  is the number of observations of the  $g$ th component of the  $\alpha$ th population. The covariance matrix  $W_g$  is a positive definite in the parameter space,  $\Omega$ .

Let  $S_g$  be the unbiased estimate of  $W_g$ , based on  $n_g$  degrees of freedom, where  $n_g = N_g - 1$  for the usual case of a random sample of  $N_g$  observation vectors from the  $g$ th population. While  $H_0$  is true,

$$S = \frac{1}{\sum n_{\alpha}} \sum_{\alpha=1}^2 n_{\alpha} S_{\alpha} \tag{38}$$

is the pooled estimate of the covariance matrices.

If all  $n_i$ s (the number of pixels in the query and target images) are equal, then  $C^{-1}$  can be written as follows:

$$C^{-1} = 1 - \frac{(2g^2 + 3g - 1)(\alpha + 1)}{6(g + 1)\alpha(N - 1)} \tag{39}$$

In the proposed work, there are only two populations, such as key face image and target face image ( $\alpha = 2$ ), and each population consists of twenty FVs ( $g = 20$ ) as depicted in Eqs. (35) and (45). The objective of this study is to test the null hypothesis that  $H_0 : W_k = W_{tr}$ , where  $W_k$  and  $W_{tr}$  represent covariance matrices of the key and target face images. The pooled estimate of the common covariance matrix,  $S$ , can be calculated as follows:

$$S = \left( \frac{1}{(N_k - 1) + (N_{tr} - 1)} \right) [(N_k - 1)S_k + (N_{tr} - 1)S_{tr}] \tag{40}$$

The test statistic for the case of two populations is given by

$$D_{Cov} = \{ [(N_k - 1) + (N_{tr} - 1)] \ln|S| - [(N_k - 1) \ln(S_k) + (N_{tr} - 1) \ln(S_{tr})] \} C^{-1} \tag{41}$$

*Critical region* If  $D_{Cov} \leq \chi_v^2(\alpha)$ , then  $H_0$  is accepted. It means that the covariance matrices of the key and target face images are closely related, that is, the key and target face images are the same or similar with respect to covariance. Otherwise,  $H_0$  is rejected, i.e. the distance between the covariance matrices of the key and target face images is greater than the critical value of the Chi-square distribution. The hypothesis is rejected means that the key face image does not match with the targeted face image in terms of covariance, where  $v = 1/2(\alpha - 1)g(g + 1)$ ,  $\alpha$  is the level of significance. If the hypothesis is rejected, then it drops the testing process and takes another facial image from the database.

### 5.2 Test for equality of mean vectors of key and target face images

If the key and target face images pass the above covariance test, then one can proceed to test the mean vectors of the two images, i.e. the test for equality of mean vectors; otherwise, the matching process could be stopped and proceed to the next image in the database. The test statistic for testing the mean vectors of the features of the key and target images is given in Eq. (43). The test of the hypothesis is formulated as follows:

*Tests of Hypothesis*

$H_0 : M_k = M_{tr}$  Null hypothesis

$H_a : M_k \neq M_{tr}$  Alternative hypothesis

The test statistic for testing the mean vectors of two populations, i.e. the key and target face images,

$$MV_{Mean} = \frac{N_k N_{tr}}{N_k + N_{tr}} (\bar{M}_k - \bar{M}_{tr})' S^{-1} (\bar{M}_k - \bar{M}_{tr}) \tag{42}$$

where

$$\bar{M}_k = \frac{1}{N_k} \sum_{g=1}^{N_g} f_k^g, \quad \text{and} \quad \bar{M}_{tr} = \frac{1}{N_{tr}} \sum_{g=1}^{N_g} f_{tr}^g; \tag{43}$$

which are distributed to a Chi-squared distribution with degrees of freedom  $v = 1/2(\alpha - 1)g(g + 1)$  (Shi et al. 2010).

*Critical region* If  $MV_{Mean} \leq \chi_v^2(\alpha)$ , then  $H_0$  may be accepted, that is, the key and target face images are the same. If  $MV_{Mean} > \chi_v^2(\alpha)$ , then the  $H_0$  may be rejected, i.e. the key and target face images are different.

If the key and target face images pass both tests, such as test for equality of covariance matrices and the test for equality of mean vectors, at 15% significance level, then the target face image is marked and indexed. The indexed face images are displayed to users. The significance level, 15%, is fixed after conducting rigorous experiments.

The overall functions of the proposed face recognition method were given in an algorithmic form as follows.

**Algorithm****Input:** colour key face image of size,  $M \times N$ .**Output:** identified face images.

// Phase 1: Pre-process

- 1: Input a colour key face image of size,  $M \times N$
- 2: Denoise the key face image using the technique discussed in Section 3.
- 3: Convert RGB  $\rightarrow$  HSV and RGB  $\rightarrow$  YCbCr
- 4: Formulate a hybrid colour model, HS-YCbCr.
- 5: Segregate each submodel into Shape and Texture parts

// Phase 2: Feature Extraction using BBPNN method

```

// Parameters are estimated through BBPNN model as follows.
// Initially, seed values are approximately given to parameters.
6: For  $l = H, S, Y, Cb, Cr$  do // First processes the submodel, H, of size,  $M \times N$ 
7:   For  $j = 1$  to  $M - 1$  // Row index of the submodel H
8:     For  $k = 1$  to  $N - 1$  // Column index of the submodel H
9:       For  $l = j$  to  $j + 2$  // Row index of the window of size  $3 \times 3$ 
10:        For  $ll = k$  to  $k + 2$  // Column index of the window of size  $3 \times 3$ 
11:         Take a window,  $w[k, l]$ , of size,  $3 \times 3$ 
12:         Next  $ll$ 
13:       Next  $l$ 
14:       Send  $w[k, l]$  and seed values to HL-1 // HL stands for Hidden Layer

15:       HL-1 | Estimate  $\hat{\alpha}$ , based on the expression in Eq. (14) with the prior-information of  $a$  received from input layer.
16:         Compute convolution function,  $g$ , in Eq. (16)
17:         If  $R \in [0.95, 1.05]$  goto Step 29 of HL-4
18:         else send  $\hat{\alpha}$  to HL-2

19:       HL-2 | Estimate  $\hat{\theta}$  and  $\hat{\phi}$ , based on the expressions in Eq. (19), using the prior-information
20:         of  $\theta$  and  $\phi$  received from the input layer and  $\hat{\alpha}$  from HL-1.
21:         Compute convolution function,  $g$ , in Eq. (16)
22:         If  $R \in [0.95, 1.05]$  goto Step 28 of HL-4
23:         else send  $\hat{\alpha}, \hat{\theta}$  and  $\hat{\phi}$  to HL-3

24:       HL-3 | Estimate  $\hat{K}$ , based on the expressions in Eq. (20), using  $\hat{\alpha}, \hat{\theta}$  and  $\hat{\phi}$  that received from HL-2.
25:         Compute convolution function,  $g$ , in Eq. (16)
26:         If  $R \in [0.95, 1.05]$  goto Step 28 of HL-4
27:         else update parameters  $\alpha = \hat{\alpha} / 2$ ;  $\theta = \hat{\theta} / 2$ ;  $\phi = \hat{\phi} / 2$ ;  $K = \hat{K} / 2$  and goto Step 15

28:       HL-4 | Compute  $\lambda_1$  and  $\lambda_2$  by substituting  $\hat{\alpha}, \hat{\theta}, \hat{\phi}$ , and  $\hat{K}$  in Eq. (18)
29:         Compute  $\rho_1$  and  $\rho_2$  by substituting  $\lambda_1$  and  $\lambda_2$  in Eqs. (29) and (30)

30:       HL-5 | Compute  $AC_m$  using Eqs. (31) and (32)
31:         If  $AC_m \in [\pm a(a' \sqrt{n})]$  then formulate  $FV^{ks}$  as in Eq. (33) //Submodel-wise
32:         else formulate  $FV^{kt}$  as in Eq. (34) //Submodel-wise
33:         Reset the parameters to  $\alpha = \hat{\alpha} / 2$ ;  $\theta = \hat{\theta} / 2$ ;  $\phi = \hat{\phi} / 2$ ;  $K = \hat{K} / 2$ 
34:         Next  $k$  (goto Step 8) // Moves to next sliding window column-wise
35:         Next  $j$  (goto Step 7) // Moves to next sliding window row-wise

36:     O/P Layer | Formulate a single FVs,  $FV^{ks}$  and  $FV^{kt}$ , for key face image as in Eqs. (33) and (34).
37:       Next  $l$  (goto Step 6) // Process another submodel
38:     O/P Layer | Combine  $FV^{ks}$  and  $FV^{kt}$  into a holistic FV,  $FV^k$ , for key face image as in Eq. (35).

```

// Phase 3: Searching and Matching

- 39: Compute  $D_{Cov}$  using Eq. (41)
- 40: If  $D_{Cov} \leq \chi_v^2(\alpha)$  then accept  $H_\theta$  and goto Step 42
- 41: else take next  $FV^r$  from FV database and goto Step 40
- 42: Compute  $MV_{Mean}$  using Eq. (42)
- 43: If  $MV_{Mean} \leq \chi_v^2(\alpha)$  then accept  $H_\theta$  and goto Step 45
- 44: else take next  $FV^r$  from FV database and goto Step 40
- 45: Infer that  $FV^k$  matches to  $FV^r$  (the key and target face images are same or similar)
- 46: Recognize and deliver the face image
- 47: End the process

## 6 Experiments and results

This section deals with databases construction, performance measures, and experimental results and discussion.

### 6.1 Experimental setups

#### 6.1.1 Face image datasets and feature databases construction

Vintra, a Californian analytics specialist and IT service provider, reported that most of the Western-based face recognition algorithms are racially biased since they have been formulated based on the publicly accessible databases that contain “supermajority” of white faces. Also, it reported that by deploying a tweak in algorithmic formulation alone could not address this problem (News 2020b).

To validate the proposed face recognition method, taking into account of the above report, we have constructed a face image database on our interest, and its details are given below. Also, we have considered the datasets, such as GT dataset (Mahmoodi and Sayedi 2015; Muqet and Holambe 2019), LFW dataset (Mahmoodi and Sayedi 2015), Pointing’04 dataset (Dantcheva et al. 2018), BioID dataset (BioID) for the experiments.

**GT dataset** Georgia Tech faces dataset composed of 750 colour face images of 50 subjects. All subjects in the dataset are represented by 15 colour JPEG images with the cluttered background taken at resolution  $640 \times 480$  pixels. The average size of the faces in these images is  $150 \times 150$  pixels. The pictures show frontal and tilted faces with different facial expressions, lighting conditions and scale. Each image is manually labelled to determine the position of the face in the image.

**LFW dataset** The LFW dataset is one of the most popular datasets in face recognition domain that comprises all variations of the face image in real-world situations, including significant variations in head pose, lighting, and skin colour. The LFW consists of 13,233 face images from 5749 different subjects like different ages, genders.

**Pointing’04 dataset** This dataset contains 15 subjects with two series of 93 images of the same subject at different head poses. The face images of the 15 subjects are with and without wearing glasses, and the skin colour may vary one subject to another. The head pose orientation is determined by two directions (horizontal and vertical) that range from  $-90^\circ$  to  $+90^\circ$ .

**BioID face dataset** (BioID, <http://www.bioid.com/downloads/facedb/index.php>) It is composed of 1521 greyscale images of 23 persons attributed to various illuminations, expression, pose variations and partial

occlusions. The face area is located in different positions with a complex background of size,  $384 \times 286$  pixels.

**Our dataset** It is composed of 1503 face images of the celebrities that gathered from Google through the Internet and many different face images captured by a digital camera.

At first, the face objects were cropped from the face datasets and pre-processed, i.e. noises removed. The noises were removed or alleviated by employing the method discussed in Sect. 3. The noise removal process is repeated until the process saturates; saturate means that the threshold ( $T$ ) falls within the band given in Eq. (44).

$$T = \frac{\text{CoV}_{t_{n-1}}}{\text{CoV}_{t_n}} \leq [0.95, 1.05] \quad (44)$$

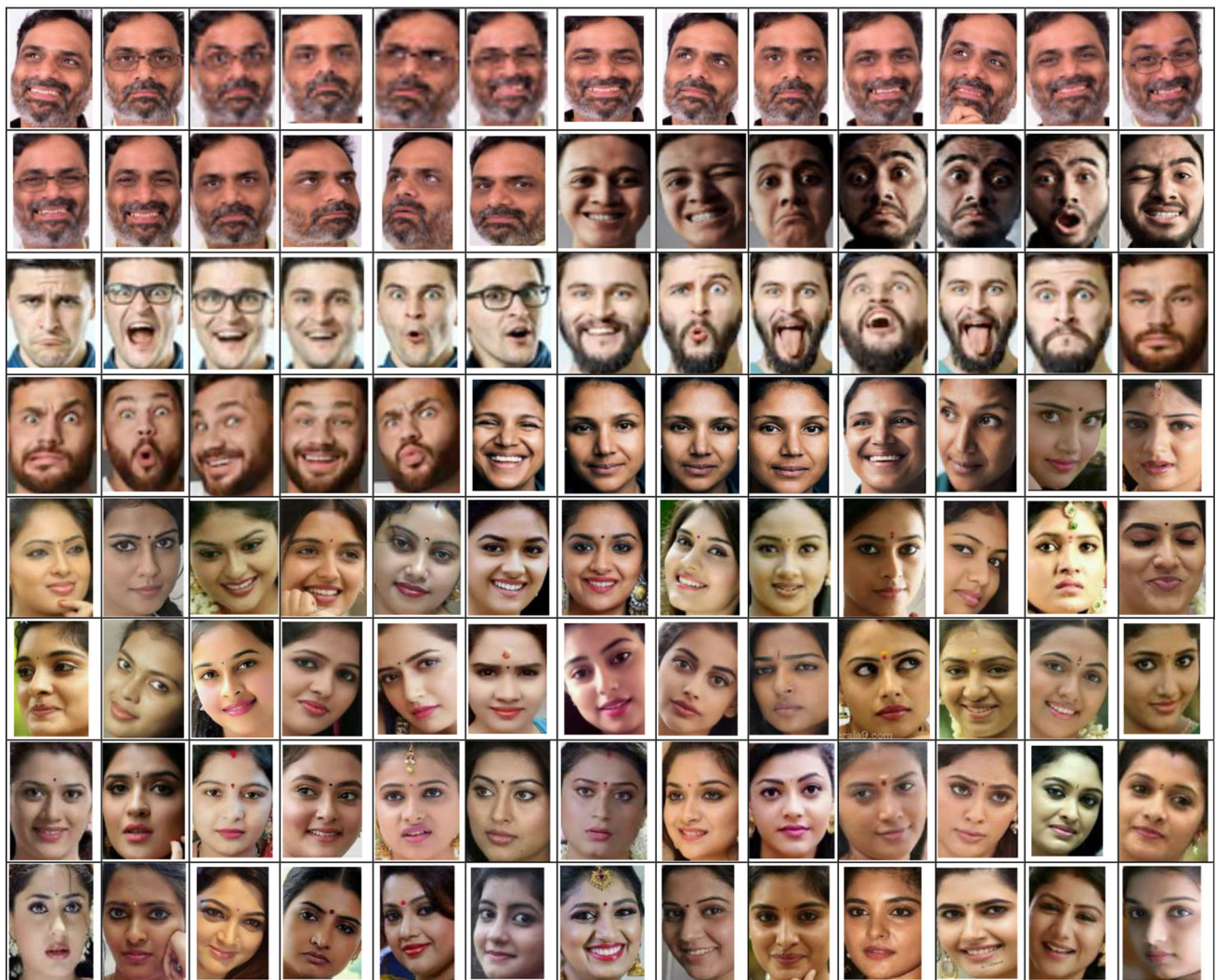
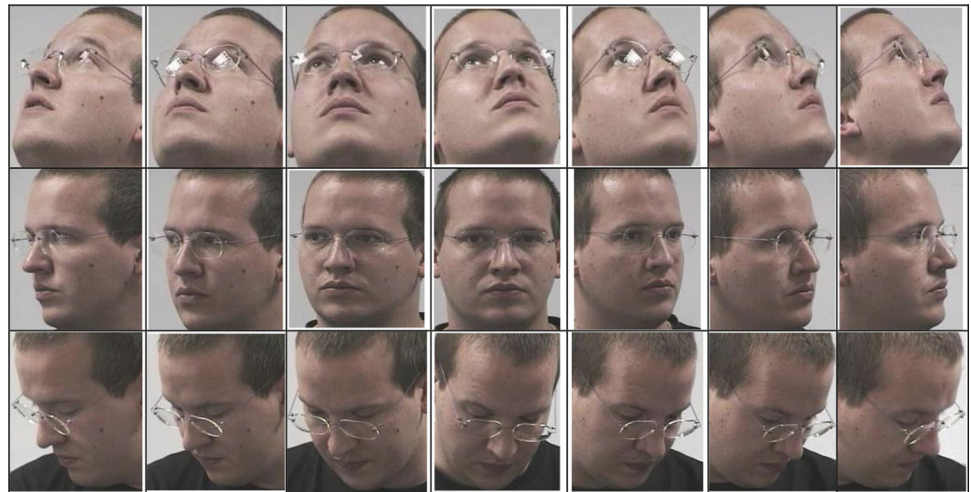
where  $\text{CoV}_{(\cdot)} = (\sigma/\mu) \times 100$ ,  $\text{CoV}_{t_{n-1}}$  and  $\text{CoV}_{t_n}$  denote the coefficient of variation for the  $(n-1)$ th and  $n$ th time process of the input face image.

The pre-processed face images were converted to HS–YCbCr colour model, and the features extracted by deploying the techniques discussed in Sect. 4, as did for the key-face image. A face FV database was constructed as discussed in Seetharaman and Jeyakarthic (2014), and the extracted face features have been stored in the feature database. The structure of the FV in the FV database is demonstrated in Eq. (45). The face features were clustered into various homogeneous groups using the *fuzzy c-means clustering algorithm* (Bezdek 1981). The median value was computed for each cluster. The FVs were indexed, based on the median value. A link was established in a one-to-one correspondence between the cluster index of the FV database and the corresponding actual face images in the database. It facilitates the users for quick recognition.

$$\overrightarrow{\text{FV}}^{tr} = (\rho_{1i}^{\text{Hs}}, \rho_{2i}^{\text{Hs}}, \rho_{1i}^{\text{Ss}}, \rho_{2i}^{\text{Ss}}, \rho_{1i}^{\text{Ys}}, \rho_{2i}^{\text{Ys}}, \rho_{1i}^{\text{Cbs}}, \rho_{2i}^{\text{Cbs}}, \rho_{1i}^{\text{Crs}}, \rho_{2i}^{\text{Crs}}, \rho_{1i}^{\text{Ht}}, \rho_{2i}^{\text{Ht}}, \rho_{1i}^{\text{St}}, \rho_{2i}^{\text{St}}, \rho_{1i}^{\text{Yt}}, \rho_{2i}^{\text{Yt}}, \rho_{1i}^{\text{Cbt}}, \rho_{2i}^{\text{Cbt}}, \rho_{1i}^{\text{Crt}}, \rho_{2i}^{\text{Crt}}) \quad (45)$$

In order to validate, the proposed ARBBPNN method is robust to head pose, and a number of images with different head poses such as the head pose ranges from  $-90^\circ$  to  $+90^\circ$  (downward to upward direction) with the interval of  $15^\circ$  as well as the head pose ranges from  $-90^\circ$  to  $+90^\circ$  (left to right at each  $15^\circ$  of downward to upward direction), collected from the database proposed in Gourier et al. (2004). Also, several face images that attribute to different facial expressions of the subjects, have been inducted in the face image database. For a sample, a few of them are presented in Figs. 4 and 5. These face images, also, have been incorporated in the newly constructed image database. The newly built face image database contains 2173 images. However, only few of them are presented in Fig. 4 owing to the space constraint.

**Fig. 4** Pointing'04 dataset. Face images with different angles



**Fig. 5** Face images with different expressions of the same person and other similar images of different persons (celebrities face images collected from Google)

### 6.1.2 System specification

The proposed method was implemented using *Python cv2* with the system specification: Intel Core i5 processor-based PC with 3.5 GHz, 16 GB DDR4 RAM, Asus Mother Board D97, and 4.0 GB Video Card.

### 6.2 Performance measure

To validate the performance of the ARBBPNN method, the *Precision@ $\alpha$*  ( $P@ \alpha$ ) was deployed, where  $\alpha$  denotes the significance level. The  $P@ \alpha$  is computed for the entire dataset as follows:

$$P@ \alpha = \frac{1}{m} \sum_{C=1}^m \left( \frac{1}{n} \sum_{g=1}^n P_g \right) \quad (46)$$

where

$$P = \frac{\text{Number of Similar Images from the Identified images}}{\text{Total Number of Images Identified}}$$

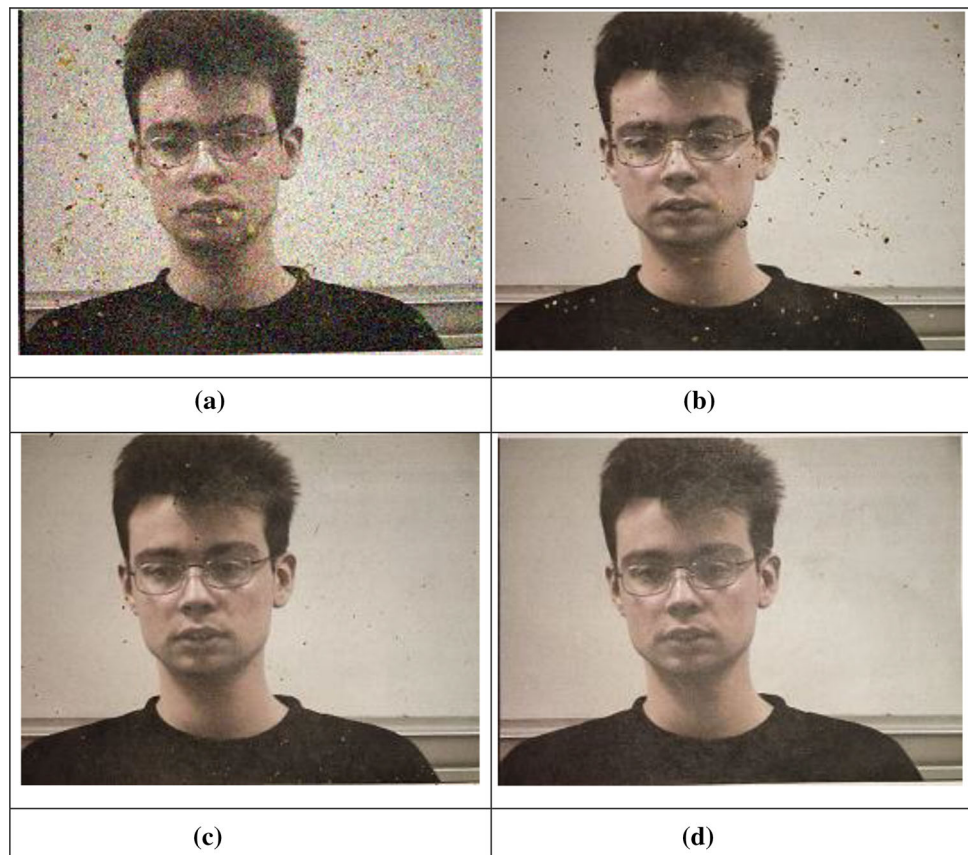
$g$  represents the number of head poses and facial expressions;  $C$  represents the number of persons in a dataset;  $\alpha$  ranges from 1 to 25% at the interval of 5%. As the precision, recall, and F1 score are most popular performance

measures in the domain of pattern recognition and information retrieval, those are not discussed in this paper.

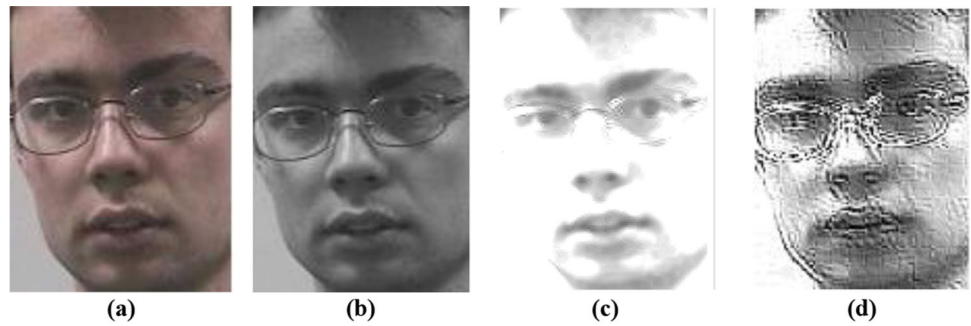
### 6.3 Experimental results

The constructed face image database is subjected to experiment to validate the proposed ARBBPNN method. The face image in Fig. 6a was submitted to the proposed method. At first, the input colour key face image was denoised as discussed earlier, based on the technique briefed in Sect. 3. For example, the pre-processed image is given in Fig. 6. In sequel, the given key image was converted to HS–YCbCr colour model. The ACC was computed from each colour component—H, S, Y, Cb, Cr, based on the sliding window of size  $3 \times 3$ . The computed ACC was tested, whether it is highly correlated or not, using the significance test discussed in Sect. 4. If the ACC is highly correlated, then it is taken into account of texture features; otherwise, it treated as shape features. For example, the shape and texture parts of an image are shown in Fig. 7. The process was repeated for the entire input image. The extracted shape and texture features were stored in a two-dimensional array, which is treated as an  $n$ -dimensional Euclidean space with Gaussian enfoldment. The shape and texture features derived from each HS–YCbCr colour

**Fig. 6** Pre-processed images.  
**a** Noised input image;  
**b** denoised image at threshold  $t_1$  (first-time); **c** denoised image at threshold  $t_2$  (second-time); denoised image at threshold  $t_3$  (third-time)



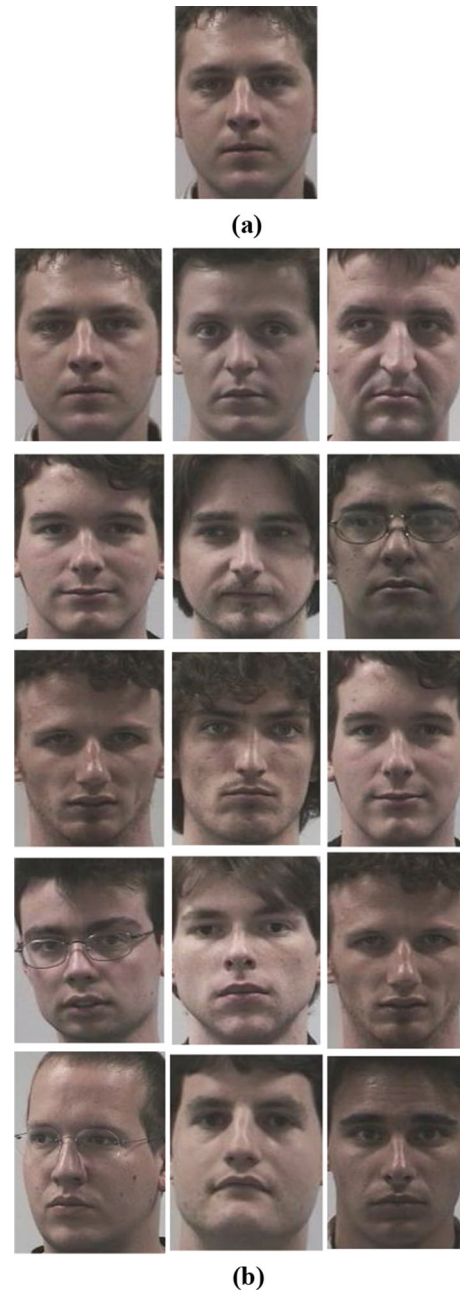
**Fig. 7** **a** Actual colour image; **b** greyscale image; **c** shape pattern; **d** texture pattern



model were formulated as a FV as described in Eqs. (33) and (34). Both shape and texture FVs were combined. As a result, there are twenty feature attributes to each key and target face images that are presented in Eqs. (35) and (45). The test statistics discussed in Sect. 5 were deployed at different significant levels. The image in Fig. 8a was submitted as an input key to the proposed system, and the identified same or similar images are shown in Fig. 8b. Images in row 1 of Fig. 8b were recognized at the 5% level of significance; images in rows 1 and 2 were identified at the 10% level of significance; at the level of significance 15%, the images in rows 1, 2, and 3 were recognized, while fixing  $\alpha$  at 20%, the images in rows 1, 2, 3, and 4 were recognized. All images in Fig. 8b were recognized while fixing  $\alpha$  at 25%.

We have made an attempt to examine the robustness of the proposed method for rotation and scale. The face images with different head poses of the fifteen persons of the Pointing'04 dataset were merged as a grand dataset and subjected to the experiments. More than fifty per cent of the head pose images were randomly selected from each subject, and the experiment was conducted at various significant levels. However, for example, only few of them have been presented owing to the space constraint. The image in Fig. 9a was submitted to the ARBBPNN system, and the corresponding identified same or similar images are presented in Fig. 9b. The images in row 1 were detected at the 5% level of significance; at 10%, the system has resulted in the images in rows 1 and 2; at 15%, the images in rows 1, 2, and 3 were identified; at 20%, the images in rows 1–4 were identified. The images in Fig. 9b were recognized while fixing  $\alpha$  at 25%.

Furthermore, the image dataset, which has face images with different expressions of South Asian people, was subjected to the experiment to examine the robustness of the proposed system for different facial expressions of a person. The image in Fig. 10a was subjected to experiments, and the system recognized the same or similar images that are given in Fig. 10b. The images in row 1 of Fig. 10b were detected at 5% level of significance; the images in rows 1 and 2 were identified at the level of



**Fig. 8** **a** Input key image; **b** identified images





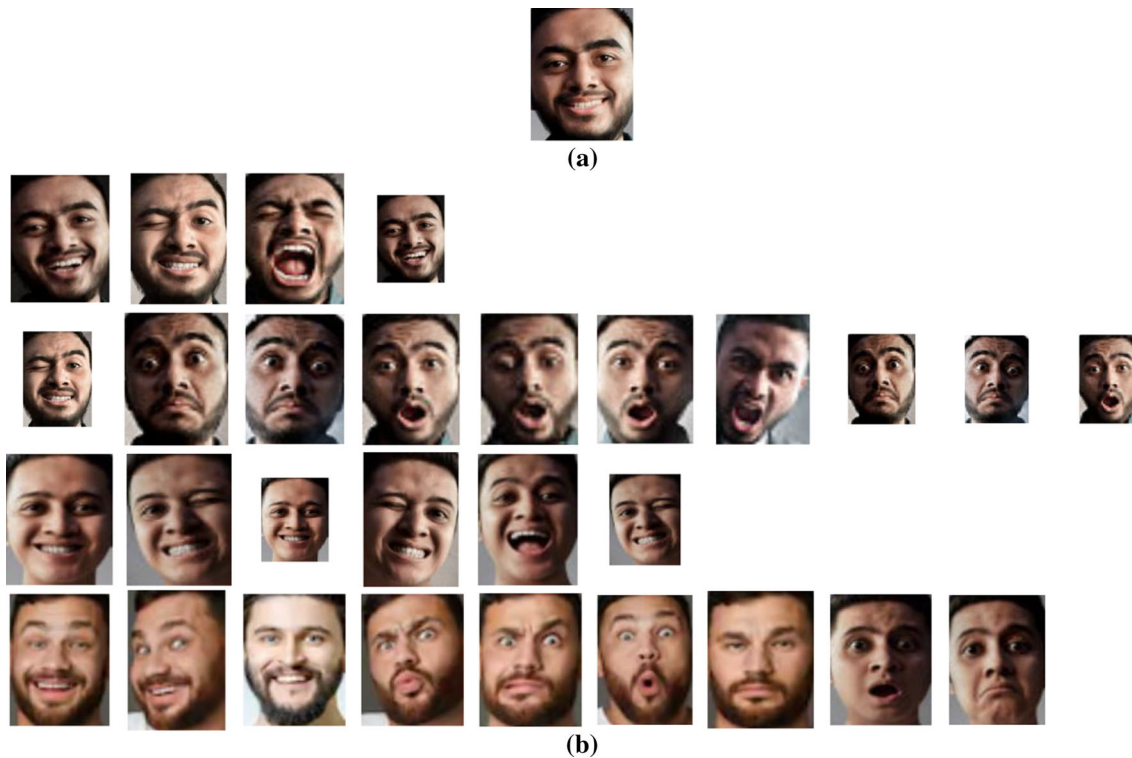
**Fig. 9** Robustness for head pose, rotation, and scale. **a** Input key image; **b** identified output images

significance, 10%. The system resulted in no images at the level of significance, 15%, while fixing the significance level at 20%, it detected the images in rows 1, 2, and 3; the images in Fig. 10b were detected at the level of significance, 25%. The experimental results show that the proposed system addresses the racial biasedness suggested in News (2020b), i.e. performs well for all types of datasets (white faces, black faces, and the trade-off between them—South Asians).

The average recognition rate was computed for the results obtained, at a particular significance level, for a subject with various combinations of the face images within the database; various combination means the combination of different head poses and facial expressions of a subject. The obtained recognition rate for each dataset is tabulated in Table 1, and a line graph was drawn, which is illustrated in Fig. 11. The results in Table 1 reveal that the proposed method shows a cent per cent average recognition rate for all the datasets at the 1% level of significance,

except the BioID dataset. At 5%, it resulted in 98.7% and 80.1% for LFW and BioID datasets, respectively, whereas it yields almost cent per cent for other datasets. At 10%, the ARBBPNN method has resulted in about cent per cent for the GT, Pointing'04, and Our datasets, while showing 97.4% and 73.81% for LFW and BioID datasets, respectively, while fixing  $\alpha$  at 15%, the ARBBPNN method has yielded 99%, 97.1%, and 97% recognition rate for Pointing'04, GT, and Our datasets, respectively, and it gives 96% and 64.3% for the LFW and BioID datasets, respectively. At the 20% level of significance, it has responded approximately 97%, 96% %, 94.4%, and 94% recognition rate for Pointing'04, GT, Our, and LFW datasets, respectively, while it performed about 58.24% for BioID dataset. The proposed ARBBPNN method has reported 96.5%, 94%, 93%, 91%, and 56% for the Pointing'04, GT, Our, and LFW datasets, respectively.

Furthermore, precision, recall, and F1 score were calculated dataset-wise, which show that the average precision



**Fig. 10** Robustness of the proposed system for different expressions of a person. **a** Input key image; **b** detected images

**Table 1** Dataset-wise  $P@z$

Sig. level	GT	LFW	Pointing'04	BioID	Our database
0.01	1	1	1	0.8825	1
0.05	1	0.9867	1	0.8014	0.9999
0.1	0.9986	0.9742	0.9999	0.7381	0.9864
0.15	0.9712	0.9597	0.9889	0.6425	0.9691
0.2	0.9587	0.9387	0.9705	0.5824	0.9441
0.25	0.9392	0.9092	0.9647	0.5604	0.9281
Average	0.9780	0.9614	0.9873	0.7012	0.9713

score for LFW and BioID datasets is 95.8% and 68.5% while around 97% for other datasets. The ARBBPNN method returns a recall score of approximately 99% for Pointing'04 dataset and 98% for GT and Our datasets; and returns 97.2% and 74.4% for LFW and BioID datasets, respectively. The ARBBPNN method resulted in about 96.5% and 71.4% F1 score for LFW and BioID datasets, respectively, while it shows about 98% score for other datasets. The dataset-wise comparison results show that the ARBBPNN method performs better for GT, Pointing'04, and proposed datasets than the LFW and BioID datasets. The following could be the main reasons:

#### Reasons for LFW dataset

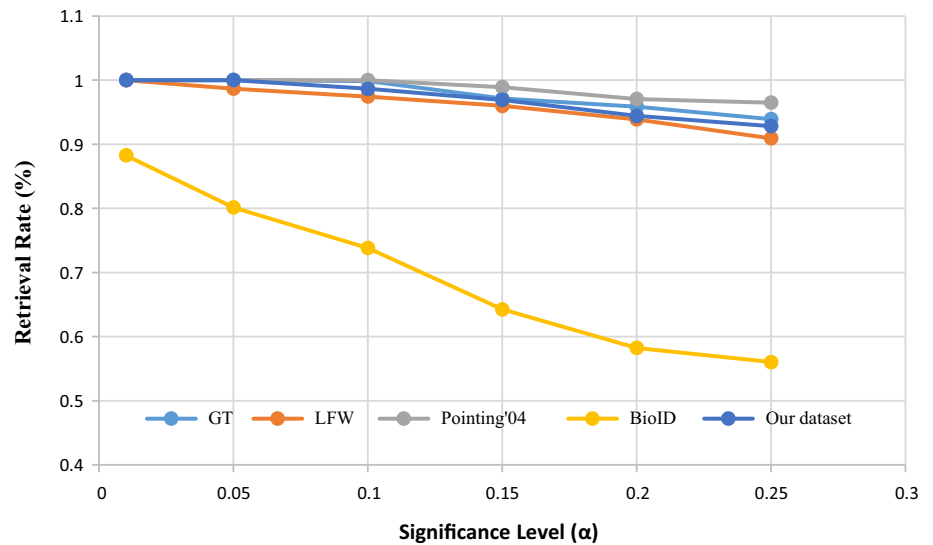
1. The size of the LFW dataset is comparatively more massive than the others.
2. Several face images suffered from poor lighting, extreme pose, strong occlusions, and low resolution.
3. The LFW dataset is not designed to have enough statistical conclusions about the subgroups.
4. About 1680 of the subjects have two or more different facial images in the data set. It reflects in the results of the *false negative* case, which also influences the F1 score.

The above complications are not prevalent in the GT, Pointing'04 datasets; especially, the Pointing'04 dataset is perfectly designed according to the persons. There also exists a problem in BioID dataset, which is different from the LFW dataset, as given below. Anyhow, the proposed ARBBPNN method has to be improved in the future to tackle the above problem even though it is not inferior to state-of-the-art methods, and it requires minimal time compared to the existing methods. The obtained results have been presented in Table 2, and a bar chart was drawn for the results in Table 2, which is depicted in Fig. 12.

#### Reason for BioID dataset

1. The features, such as shape and texture, are extracted from the greyscale image only. Hence, totally, only four features out of twenty are extracted for the BioID

**Fig. 11** Line graph for different datasets: significance level versus retrieval rate



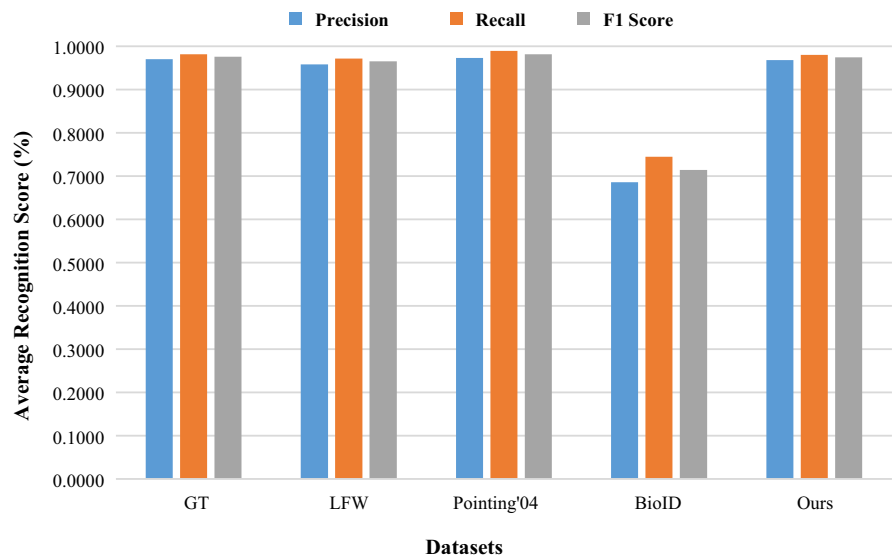
**Table 2** Dataset-wise performance measure

Dataset	Precision	Recall	F1 score
GT	0.9701	0.9812	0.9756
LFW	0.9582	0.9715	0.9648
Pointing'04	0.9732	0.9892	0.9811
BioID	0.6852	0.7443	0.7139
Our dataset	0.9679	0.9801	0.9740

dataset. It is minimal while compared to that of the other datasets. It evidences that the HS-YCbCr colour model and the features extracted from it are more appropriate and sufficient for achieving high accuracy of face recognition.

Moreover, to validate the proposed ARBBPNN method, the obtained results were compared to state-of-the-art methods in terms of F1 score. The F1 score obtained by the ARBBPNN method and the F1 scores available in Mahmoodi and Sayedi (2015), Muqet and Holambe (2019) and Liu et al. (2019) have been taken into account for comparison, which is given in Table 3. In Mahmoodi and Sayedi (2015), the F1 score is available only for the GT and LFW datasets; in CRRF (Liu et al. 2019) method, the F1 score was calculated for LFW dataset; we calculated the F1 score for the DIWT method using the rank-one score and other results available in Muqet and Holambe (2019). The results presented in Table 3 show that the proposed method gives better results than the state-of-the-art methods in terms of F1 score. Figure 13 illustrates the results given in Table 3.

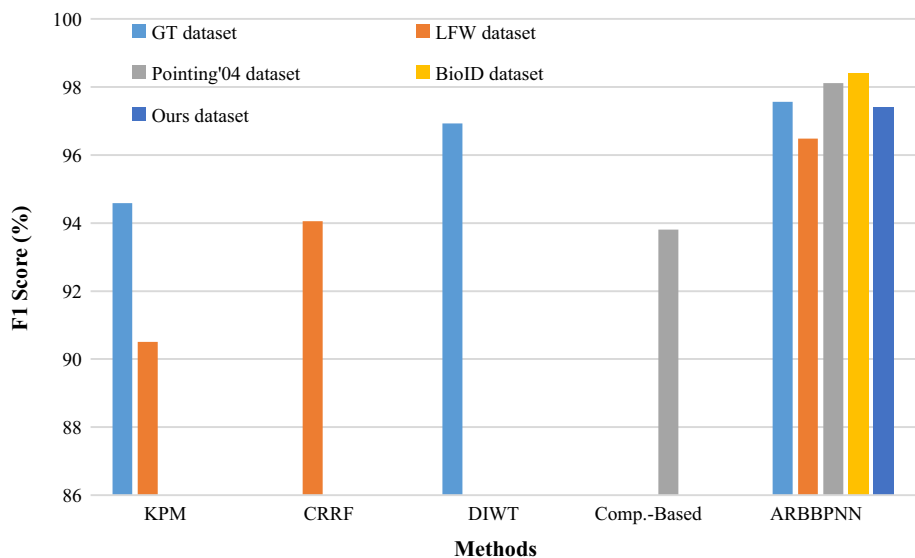
**Fig. 12** Dataset-wise performance measures



**Table 3** Comparison of different methods in terms of F1 score

Method	Dataset				
	GT	LFW	Pointing'04	BioID	Our
KPM (Mahmoodi and Sayedi 2015)	94.59	90.51	–	–	–
CRRF (Liu et al. 2019)	–	94.05	–	–	–
DIWT (Muqet and Holambe 2019)	96.93	–	–	–	–
Component-based method (Paul et al. 2020)	–	–	–	47.04	–
Proposed ARBBPNN	97.56	96.48	98.11	71.39	97.40

**Fig. 13** Comparison of performance of different methods in terms of F1 score



**Table 4** Dataset-wise computational cost for the proposed ARBBPNN method

Phases of process	Average time consumed (in seconds)				
	GT	LFW	Pointing'04	BioID	Our
Pre-processing the key face image	16	17	15	09	17
BBPNN method for feature extraction	45	46	46	22	46
Searching and matching	16	42	21	12	23
Holistic face recognition process	77	105	82	43	86

### 7 Computational cost

In literature, most articles discuss the computational cost in terms of qualitative rather than quantitative (Muqet and Holambe 2019; Paul et al. 2020). Therefore, this paper presents the computational cost for only the proposed ARBBPNN method. The computation cost was measured from submission of the input key face image until obtaining the output recognized face images. The computational cost was calculated phase-wise, and it is given in Table 4.

The obtained results show that the overall time consumed for BioID dataset was lesser than the others because

it has only one component, viz. the greyscale ( $Y$  submodel), while the others have five submodels. It is the main reason that only four feature attributes, such as  $Y^{rr}(\rho_1, \rho_2)$  and  $Y^{rrs}(\rho_1, \rho_2)$ , are extracted from the BioID dataset so that it consumes lesser time for feature extraction. It shows the insufficiency of the features, which has reflected in the accuracy of the recognition rate. The obtained recognition results are presented in Table 3. The other datasets, except LFW, demand almost the same computational cost. Because of the LFW dataset is composed of 13,233 face images, which is larger while compared to others, it demands more time than the others.

## 8 Discussion and conclusion

### 8.1 Discussion

The proposed ARBBPNN method achieves good performance for biometric-based face image detection, which outperforms the methods developed, based on the hand-crafted features; also, it is comparable to the systems developed with the deep neural network. The ARBBPNN method acts as a trade-off between the deep learning methods and the hand-crafted methods in terms of less computational cost and high accuracy. Because the ARBBPNN method employs BBPNN method for only the features extraction and adopts hand-crafted techniques for feature matching, it has the properties of both deep learning and hand-crafted methods. At the same time, it maintains high accuracy, as the feature matching is performed in two stages: (1) test the variation between the key and target face images with respect to covariance; (2) test the equality of spectrum of FVs of the key and target face images, provided the images pass the test at stage (1). Though the performance of the ARBBPNN method is comparable to the deep neural network-based methods, we have a plan, in the future, to improve the ARBBPNN method in the light of the end-to-end deep learning methods.

The precision and F1 score obtained dataset-wise evidence that the HS–YCbCr colour model is sufficient and efficient for face characterization and feature extraction than the single colour model like greyscale.

### 8.2 Conclusion

The proposed ARBBPNN method was developed for biometric-based face recognition because of the sudden demand for non-touch biometric systems like face recognition in the future due to the impact of pandemic disease COVID-19. Both public and private organizations that maintain a biometric-based attendance system will begin to shun the touch-based methods like a fingerprint. The proposed system is a hybrid of the hand-crafted, and DNN features so that it results in a reasonable face detection accuracy with minimal computational cost. The obtained results are comparable to state-of-the-art methods. We have subjected four different publicly available benchmark datasets—GT, LFW, BioID, and Pointing'04—to the experiments. In addition to that, we constructed a dataset with celebrities' face images, which is also subjected to the experiments. The proposed ARBBPNN-based method has responded around 97.5% average recognition rate for GT and Our datasets, while 98.1%, 96.5%, and 71.4% for Pointing'04, LFW, and BioID datasets, respectively. The

obtained average recognition results are comparable to state-of-the-art methods.

### 8.3 Future direction

The accuracy of the recognition rate of the ARBBPNN method can be improved in the future, even for the datasets that have complicated nature or structures like LFW and other datasets with massive facial images, in the light of the end-to-end DNN and higher-order AR model. The proposed ARBBPNN method could be deployed for iris recognition and also in the domain of big-data analytics.

**Funding** This study was not funded by any funding agency.

### Compliance with ethical standards

**Conflict of interest** The authors M. Vasanthi and K. Seetharaman do not have any conflict of interest.

**Human and animal rights** As this study was fully involved in mathematical and computational studies, there are no possibilities of use of human or animals for experimental purposes. This article does not contain any studies with human participants or animals performed by any of the authors.

## References

- Anderson TW (2003) An introduction to multivariate statistical analysis, 3rd edn. Wiley, New York
- Arashloo SR (2016) A comparison of deep multilayer networks and Markov random field matching models for face recognition in the wild. *IET Comput Vis* 10(6):466–474
- Bezdek JC (1981) Pattern recognition with fuzzy objective function algorithms. Plenum Press, New York
- Bindemann M, Burton M (2009) The role of color in human face detection. *Cogn Sci* 33:1144–1156
- BioID Face Database downloaded from <http://www.bioid.com/downloads/facedb/index.php>
- Carbó-Dorca R, Besalú E (2011) Geometry of n-dimensional Euclidean space Gaussian enfoldments. *J Math Chem* 49:2244
- Carlaw S (2020) Impact on biometrics of Covid-19. *Biom Technol Today* 2020(4):8–9
- Chai Z, Sun Z, Méndez-Vázquez H, He R, Tan T (2014) Gabor ordinal measures for face recognition. *IEEE Trans Inf Forensics Secur* 9(1):14–26
- Dantcheva A, Bilinski P, Bremond F (2018) Show me your face and I will tell you your height, weight and body mass index. In: Proceedings of 24th IAPR international conference on pattern recognition (ICPR), August 2018, Beijing, China
- Goswami G, Agarwal A, Ratha N, Singh R, Vatsa M (2019) Detecting and mitigating adversarial perturbations for robust face recognition. *Int J Comput Vis* 127:719–742
- Gourier N, Hall D, Crowley JL (2004) Estimating face orientation from robust detection of salient facial features. In: Proceedings of pointing 2004, ICPR, international workshop on visual observation of deictic gestures, Cambridge, UK. <http://www.prima.inrialpes.fr/Pointing04/>

- Guo Z, Zhang L, Zhang D (2010) A completed modeling of local binary pattern operator for texture classification. *IEEE Trans Image Process* 19(6):1657–1663
- Hatamikia S, Maghooli K, Nasrabadi AM (2014) The emotion recognition system based on autoregressive model and sequential forward feature selection of electroencephalogram signals. *J Med Signals Sens* 4(3):194–201
- He R, Tan T, Davis L, Sun Z (2018) Learning structured ordinal measures for video based face recognition. *Pattern Recognit* 75:4–14
- Kukush A (2019) Gaussian measures in Hilbert space: construction and properties. ISTE Ltd, Wiley, New York
- Liu Z, Liu C (2008) A hybrid color and frequency features method for face recognition. *IEEE Trans Image Process* 17(10):1975–1980
- Liu Z, Liu C (2010) Fusion of color, local spatial and global frequency information for face recognition. *Pattern Recognit* 43:2882–2890
- Liu B, Xiao Y, Yu PS, Hao Z, Cao L (2014) An efficient approach for outlier detection with imperfect data labels. *IEEE Trans Knowl Data Eng* 26(7):1602–1616
- Liu L, Gui W, Zhang L, Chen J (2019) Real-time pose invariant spontaneous smile detection using conditional random regression forests. *Optik* 182:647–657
- Mahmoodi MR, Sayedi SM (2015) A face detection method based on kernel probability map. *Comput Electr Eng* 46:205–206
- Mil J-Z, Liu J-X, Wen J (2012) New robust face recognition methods based on linear regression. *PLoS One* 7(8):1–10
- Muqet MA, Holambe RS (2019) Local binary patterns based on directional wavelet transform for expression and pose-invariant face recognition. *Appl Comput Inform* 15(2019):163–171
- News (2020a) Coronavirus response—face ID firms battle Covid-19 as users shun fingerprinting. *Biom Technol Today* 2020(4):1–2
- News (2020b) Facial recognition—Vintra claims less racial bias than Microsoft and Amazon. *Biom Technol Today* 2020(4):2
- Paul SK, Bouakaz S, Rahman CM, Uddin MS (2020) Component-based face recognition using statistical pattern matching analysis. *Pattern Anal Appl*. <https://doi.org/10.1007/s10044-020-00895-4>
- Pavithra LK, Sharmila S (2017) An efficient framework for image retrieval using color, texture and edge features. *Comput Electr Eng* 70:580–593
- Pena D, Rodriguez J (2002) A powerful portmanteau test of lack of fit for time series. *J Am Stat Assoc* 97(458):601–610
- Pujol FA, Pujol M, Jimeno-Morenilla A, Pujol MJ (2017) Face detection based on skin color segmentation using fuzzy entropy. *Entropy* 19(26):1–22
- Seetharaman K (2012) A block-oriented restoration in grayscale images using full range autoregressive model. *Pattern Recognit* 45(4):1591–1601
- Seetharaman K (2015) Image retrieval based on micro-level spatial structure features and content analysis using full range Gaussian Markov random field model. *Eng Appl Artif Intell* 40:103–116
- Seetharaman K, Jeyakarthic M (2014) Statistical distributional approach for scale and rotation invariant color image retrieval using multivariate parametric tests and orthogonality condition. *J Vis Commun Image Represent* 25(5):727–739
- Seetharamana K, Palanivel N (2013) Texture characterization, representation, description, and classification based on full range Gaussian Markov random field model with Bayesian approach. *Int J Image Data Fus* 4(4):342–362
- Sharifara A, Rahim MSM, Navabifar F, Ebert D, Ghaderi A, Papakostas M (2017) Enhanced facial recognition framework based on skin tone and false alarm rejection. In: *Proceedings: 10th international conference on PErvasive technologies related to assistive environments—PETRA'17*. <https://doi.org/10.1145/3056540.3064967>
- Shi Q, Li H, Shen C (2010) Rapid face recognition using hashing. In: *Proceedings: CVPR, IEEE computer society conference on computer vision and pattern recognition*
- Singh G, Chabra I (2018) Effective and fast face recognition system using complementary OCLBP and HOG feature descriptors with SVM classifier. *J Inf Technol Res* 11(1):91–110
- Tan X, Triggs B (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Trans Image Process* 19(6):1635–1650
- Terrillon JC, Shirazi MN, Fukamachi H, Akamatsu S (2000) Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In: *IEEE international conference on face and gesture recognition*, pp 54–61
- Thakur S, Paul S, Mondal A, Das S, Abraham A (2011) Face detection using skin tone segmentation. In: *World congress on information and communication technologies, Mumbai*, pp 53–60
- Theoharatos C, Laskaris NA, Economou G, Fotopoulos S (2005) A generic scheme for color image retrieval based on the multivariate Wald–Wolfowitz test. *IEEE Trans Knowl Data Eng* 17(6):808–819
- Uçar A (2014) Color face recognition based on steerable pyramid transform and extreme learning machines. *Sci World J* 2014:1–5
- Vapnik V (1998) *Statistical learning theory*. Springer, London, p 1998
- Wang R, Lei Z, Ao M, Li SZ (2009) Bayesian face recognition based on Markov random field modeling. In: *Tistarelli M, Nixon MS (eds) Advances in biometrics. ICB 2009, vol 5558. Lecture notes in computer science*. Springer, Berlin
- Wang J, Tai S, Lin C (2018) The application of an interactively recurrent self-evolving fuzzy CMAC classifier on face detection in color images. *Neural Comput Appl* 29:201–213
- Wu B, Hua B-G, Ji Q (2017) A coupled hidden markov random field model for simultaneous face clustering and tracking in videos. *Pattern Recognit* 64:361–373
- Wu S, Kan M, Shan S, Chen X (2019) Hierarchical attention for part-aware face detection. *Int J Comput Vis* 127:560–578
- Yan J, Zhang X, Lei Z, Li SZ (2013) Face detection by structural models. *Image Vis Comput* 32(10):790–799

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.