Neural Processing Letters
https://doi.org/10.1007/s11063-018-9921-6

A Comparative Study of Suitability of Certain Features in Classification of Bharatanatyam Mudra Images Using Artificial Neural Network

Basavaraj S. Anami[1] · Venkatesh A. Bhandage[2]

Abstract
Bharatanatyam is an Indian classical dance, which is composed of various body postures and hand gestures. This ancient art of dance has to be studied under the supervision of experts but at present there is dearth of Bharatanatyam dance experts. This has led to take leverage of technology to make this dance self pursuable. Thus, it is the motivation for automation of identification of mudras through image processing. This paper presents a 3-stage methodology for classification of single hand mudra images. The first stage involves acquisition and preprocessing of images of mudras to obtain contours of mudras using canny edge detector. In the second stage, the features, namely, Hu-moments, eigenvalues and intersections are extracted. In the third stage artificial neural network is used for classification of mudras. The comparative study of classification accuracies of using different features is provided at the end. To corroborate the obtained classification accuracies, a deep learning approach, namely, convolutional neural network is adopted. The work finds application in e-learning of 'Bharatanatyam' dance in particular and dances in general and automation of commentary during concerts.

Keywords Mudra · Contour of mudras · Hu-moments · Eigenvalues · Intersections · Artificial neural network · Convolutional neural network

1 Introduction

Recent advances in digital image processing have led to the development of newer applications such as hand gesture recognition, body posture recognition and human action recognition. One such significant application is seen in Bharatanatyam dance and mudra classification. Bharatanatyam dance is an ancient Indian classical dance form which is composed of various body postures, facial expressions and coordinated movement of various hand gestures which are performed to the accompaniment of dance syllables. These dance

✉ Venkatesh A. Bhandage
venkateshabhandage@gmail.com

[1] K.L.E. Institute of Technology, Hubballi, Karnataka, India

[2] Tontadarya College of Engineering, Gadag, Karnataka, India

Published online: 24 September 2018

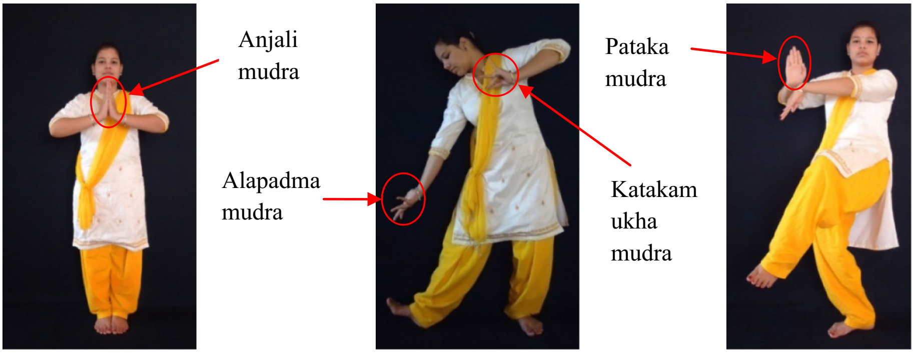


Fig. 1 Typical postures in Bharatanatyam dance with mudras

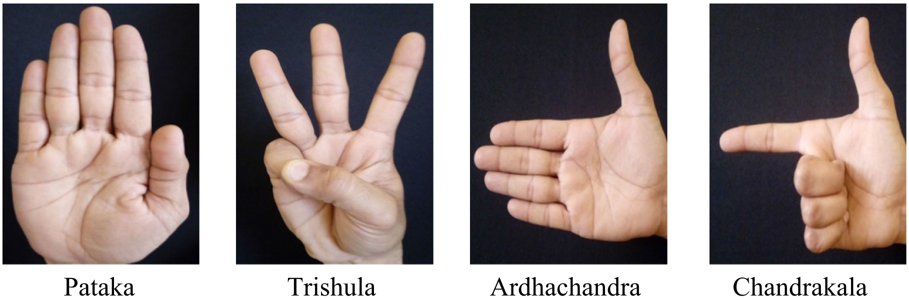


Fig. 2 Bharatanatyam dance mudras

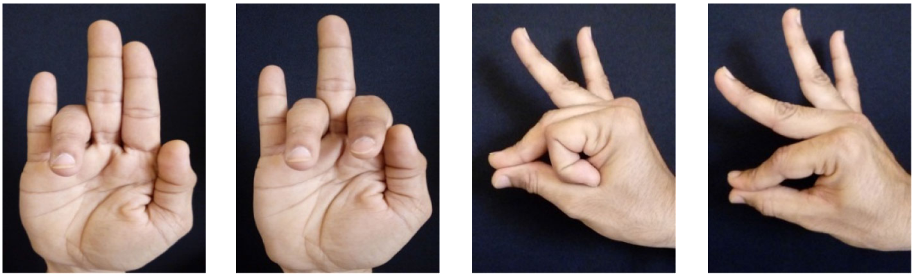
movements convey some useful meaning to the audience. The semantic meaning and feelings of these dance syllables can only be properly understood by the connoisseur. The novice learners and general audience may find it difficult to understand the inner as well as outer feelings conveyed by the dancers. The Natyashastra, an earlier repository of Indian classical dance form has details of dance syllables used in various dance forms. It mentions a set of 108 dance postures called k*aranas*, 28 *Asamyukta hastas,* 23 *Samyukta hastas and 13 Nrita hastas, or* decorative hand gestures. The 24th hasta '*Avahitta*' is recently added to the list of *Samyukta mudras.* The hasta is also called as a *mudra*. Asamyukta mudras are performed using single hand and samyukta mudras are performed using both hands. These karanas and mudras are prominent parts of the dancer's vocabulary. Figure 1 shows typical postures of some of Bharatanatyam dance along with some mudras. A mudra is the most striking feature of this Indian classical dance, which uses hand gestures. Hand sign is a ritual gesture in Hinduism and Buddhism. Some single hand mudras are shown in Fig. 2.

In this work we have focussed on developing a system for classification of asamyukta mudras. The list of asamyukta mudras is given in Table 1. In asamyukta mudras, some mudras conflict and hence the mudras are further classified into conflicting and non-conflicting mudras. Figure 2 shows some of the non-conflicting mudras. Some of the conflicting mudras are shown in Fig. 3. Tripataka mudra is conflicting with shukatunda mudra and bhramara mudra is conflicting with hamsassya mudra.

The shapes of mudras are important in their classification. Many shape descriptors have been proposed for different applications. Popularly used shape descriptors are moments

Table 1 List of asamyukta mudras

1	Pataka	8	Shukatunda	15	Padmakosha	22	Bhramara
2	Tripataka	9	Mushti	16	Sarpashirsha	23	Hamsassya
3	Ardhapataka	10	Shikhara	17	Mrugashirsha	24	Hamsapakshika
4	Kartarimukha	11	Kapitta	18	Simhamukha	25	Samdamsha
5	Mayura	12	Katakamukha	19	Kangula	26	Mukula
6	Ardhachandra	13	Suchi	20	Alapadma	27	Tamrachuda
7	Arala	14	Chandrakala	21	Chatura	28	Trishula



Tripataka Shukatunda Bhramara Hamsassya

Fig. 3 Conflicting mudras in Bharatanatyam dance

and eigenvalues. There are various types of moments cited in the literature such as, Hu-moments, Zernike moments, pseudo-Zernike moments, Tchebichef moments, orthogonal Fourier–Mellin moments (OFMMs), Complex moments, and Gaussian-Hermite moments (GHMs). Eigenvalues of a square matrix are principal components, which contain most details of the image. Moments and eigenvalues find applications in various areas like character recognition, face recognition, hand gesture recognition and facial expression recognition. It is being attempted to deploy Hu-moments, eigenvalues and intersections for the recognition and classification of Bharatanatyam mudra images.

The motivation for the proposed work is that Bharatanatyam dance and mudras are to be studied under an expert, it is becoming difficult to find experts in villages, semi-urban areas and foreign countries. Propagating this Indian art within and outside the country requires technological leveraging. Hence, automation of classification of this art of dance and mudras through image processing is being attempted and single hand mudras are considered. The work is challenging as there are mudras, which look similar with other mudras and hence may lead to misclassification of mudras. Identifying the mudras from the video or online is considered not under the purview of this paper.

The work involves obtaining contour of the input mudra, obtaining features such as Hu-moments, eigenvalues and intersections of the grid lines with the contour of mudra, use of these feature sets for recognition and classification of unknown mudras using artificial neural network classifier. To corroborate the results, we have implemented a deep learning approach, namely, the convolution neural network classifier.

The remaining part of the paper is organized into four sections. Literature survey is given in Sect. 2. The proposed methodology is given in Sect. 3. Section 4 gives comparative study. Conclusion is given in Sect. 5.

2 Literature Survey

In order to know the state-of-the-art in mudra classification and allied works, a literature survey is carried out. The gist of papers is as given under.

Anami and Bhandage [1] have proposed an approach for identification of double hand mudra images of Bharatanatyam dance. They have used vertical-horizontal-intersections and types of mudras as features. Mudras are discriminated as conflicting and non-conflicting. A rule based classifier is developed for classification of 24 classes of double hand mudra images used in Bharatanatyam dance. The reported average accuracy is 95.25% [1].

Liu et al. [2] have proposed a methodology for scene recognition using p-Laplacian regularization. They have introduced an approximation algorithm of graph p-Laplacian, which reduces the cost of computation and proposed pLapR to preserve the local similarity of data. A pLapR to support vector machines (SVMs) and kernel least squares (KLSs) for scene recognition are presented. They have experimented on the datasets Scene 67, Scene 15 and UC-Merced. It is shown that the proposed pLapR outperforms the conventional manifold regularization methods including LapR and HesR [2].

Yu et al. [3] have proposed a method for ranking images being searched. They have developed a deep multimodal distance metric learning (Deep-MDML) method. They have adopted a model, which utilizes multimodal features including visual and click features in distance metric learning (DML). They have experimented on two benchmark datasets Microsoft Bing Search Engine (LSIS) and Microsoft Image Retrieval Challenge (IRC). It is demonstrated that proposed rank model improves visual searching [3].

Kumar and Kishore [4] have presented an approach for classification of mudras of Indian classical dance. They have used HOG Features and SVM Classifier. A set of mudras present in Indian classical dance forms are considered. They have compared the recognition accuracy of the presented system by using other feature vectors such as SIFT, SURF, LBP and HAAR. The recognition accuracy by using other features is less than 80%. HOG features have produced an average accuracy of 90% [4].

Solís et al. [5] have proposed a methodology for Mexican sign language (MSL) recognition. They have used forty two normalized central moments as features. Multi-layer Perceptron (MLP) is used as a classifier. The isolated Mexican sign language alphabets are recognized with a recognition accuracy of 93% [5].

Zadghorban and Nahvi [6] have presented an algorithm for recognition of continuous Persian sign language (PSL). Hand shape features such as Fourier descriptors, Hu-moments and Zernike moments, motion dependent features such as hand motion and dynamic time warping (DTW) are used as features. They have used hidden markov model (HMM) for motion feature based classification and hybrid KNN-DTW algorithm for classification based on hand shape features. The algorithm has given an overall recognition accuracy of 93.73% [6].

Wang et al. [7] have proposed a system for hand gesture based robot movement control. The valley circle (VC) is created for the first stage of 6 fingertip numbers classification. The hybrid feature vector of Hu-moments, convexity and compactness (HCC) are constructed at the second stage. In the final stage, a new template matching recognition (NTMR) algorithm is devised for recognition of 10 gesture classes [7].

Pradhan et al. [8], have proposed a hand gesture recognition system based on principal component analysis. They have used eigenvalues and eigenvectors as features. Euclidean distance is used as a measure of similarity between known gesture and unknown gesture. They have considered a dataset of 78 images, 3 images for each alphabet. They have obtained an efficiency of 65.38% [8].

Hong et al. [9] have proposed a method for recovering 3D human poses from 2D images. They have used multimodal deep autoencoder (MDA) and non-linear backpropagation neural network (BP-NN). They have combined different types of features using MDA and they have used BP-NN to map 2D silhouettes to 3D poses. They have experimented on Walking, HumanEva and Human3.6 M datasets. They have obtained 20–25% of reduction in recovery error [9].

Fagiani et al. [10] have presented an approach for isolated Italian sign recognition. They have considered various features such as hands centroid normalized respect face region width, normalized hands centroid distance, Hu-moments of face and hand, hands area normalized respect to the face area, face and hands compactness, face and hands eccentricity, and hands orientation. They have used hidden markov model as a classifier. They have obtained an overall recognition accuracy of 48.06% [10].

Fernando and Wijayanayake [11] have proposed a methodology for real time sign language communication. They have used Hu-moments, contour, orientation histogram, convex hull, convexity defects as features. They have found that Hu-moments based classification is good. To improve the accuracy of the system, they have used height to width ratio filtration along with the Hu-moments. The system has provided a recognition accuracy of 84% [11].

Nguyen and Huynh [12], have presented an approach for recognition of static hand gestures. They have used eigenvectors as features. Artificial neural network is used for classification of unknown gestures. The system recognizes 24 gesture classes with an overall accuracy of 94.3% [12].

Dixit and Jalal [13] have presented a three phase system for recognition of Indian sign language. A combination of Hu-moment invariants and structural shape descriptors are used as features. A dataset of 720 images is considered. Multi-class support vector machine classifier is used for recognizing signs. A recognition accuracy of 96% is obtained [13].

Premaratne et al. [14] have proposed a system for recognition of Australian sign language. They have used moment invariants as features. They have used Auslan database which contains 7415 words. They have used support vector machine (SVM) and neural network (NN) as classifiers. The system is able to recognize the numbers zero to nine with occasional errors in few gestures [14].

Sriparna et al. [15] have presented an approach for hand gesture recognition of Bharatanatyam dance. They have used fuzzy L-membership function and a 3-Stage system is developed. In the first stage, contour of hand gesture is obtained using Sobel edge detection operator. In second stage, center of the contour is found and eight spatial distances are calculated. In third stage, fuzzy L-membership values are calculated for each distance and matching of unknown hand gesture is carried out. They have reported an average accuracy of 85.1% [15].

Adithya et al. [16], have presented a methodology for recognition of Indian sign language. They have used skin colour based segmentation to extract hand from the background. They have considered various shape based features for training and testing artificial neural network. They have reported an average recognition accuracy of 91.11% [16].

Singha and Das [17], have presented a system for Indian sign language recognition. They have used skin color based technique to segment the gesture portion from the hand part. They have calculated eigenvalues and eigenvectors from the gesture image. Euclidean distance is used as a measure to identify the unknown gesture image. They have considered a dataset of 240 images. An overall accuracy of 97% is reported [17].

Liu et al. [18] have proposed a methodology for hand gesture recognition for interaction in virtual reality. Skin colour detection technique is used for obtaining hand region in the image. Hu-moments of the hand gesture are used as features and support vector machine is used as a classifier [18].

Otiniano-Rodríguez et al. [19] have presented a system for sign language recognition using Hu and Zernike moments. They have considered a public database of 2040 images stating 24 symbols classes. They have presented two approaches for sign language recognition (SLR) using support vector machine as a classifier. In the first approach they have used Hu-moments as features and in the second approach they have used Zernike moments as features. Hu-moments have given an overall accuracy of 93% and Zernike moments have given an overall accuracy of 96% [19].

Hariharan et al. [20] have proposed an approach for recognition of dancer's hand gestures. The methodology recognizes single hand gestures and is scale, translation and rotation invariant. Orientation filters are used at the first level. At the second level, the contour of gesture is obtained and the gradients are calculated. These gradients are used for recognition of gestures which are unidentified at the first level. They have developed a two level system for recognition of Bharatanatyam dance gestures [20].

Zaki and Samir [21] have proposed a system for recognition of sign language. They have considered four components of sign languages such as hand shape, place of articulation, hand orientation, and movement. They have used kurtosis position, principal component analysis (PCA) and motion chain code as features and hidden markov model as a classifier. They have reported an overall error rate of 10.90% [21].

Prashant Sharma et al. [22], have presented a methodology for face recognition. They have calculated eigenvectors and eigenvalues from the initial face image with the help of distinct block processing. They have projected new faces onto the space expanded by eigenfaces and they have represented new faces by weighted sum of these eigenfaces. They have used these weights to identify the faces. They have obtained a recognition rate of 98.5% using artificial neural network.

Lecun et al. [23] have provided a review on various methods used for handwritten character recognition. They have adopted gradient-based learning for recognizing documents and shown that convolutional neural networks can be effectively trained to recognize objects directly from their images [23].

Hu [24] has presented a methodology for visual pattern recognition by moment Invariants. He has established a fundamental theorem to relate such moment invariants to the well known algebraic invariants. Complete systems of moment invariants under translation, similitude and orthogonal transformations are derived. It is shown that recognition of geometrical patterns and alphabetical characters independent of position, size and orientation are accomplished [24].

From the literature survey, it is observed that good amount of work has been carried out in sign language recognition and hand gesture recognition. Deep learning techniques are used for identifying human poses and scenes from their images. Researchers have used moments and eigenvalues as features with support vector machine and artificial neural network classifiers. Considering Bharatanatyam mudras as types of gestures or signs, a limited work is carried out on recognition of Bharatanatyam mudras' images. The reported research works have used less number of images of mudras for recognition. The accuracy reported has scope for improvement and hence the motivation for the present work.

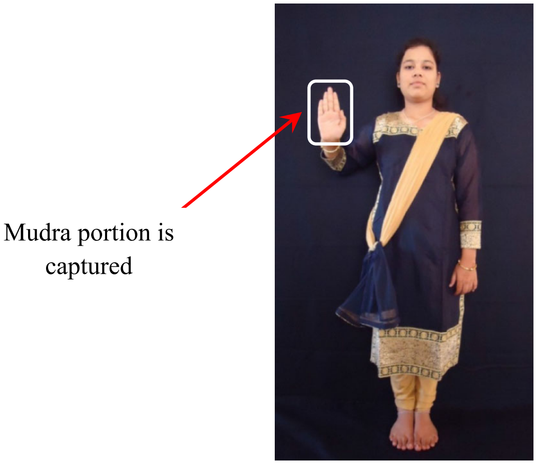


Fig. 4 Setup for capturing Bharatanatyam mudra images

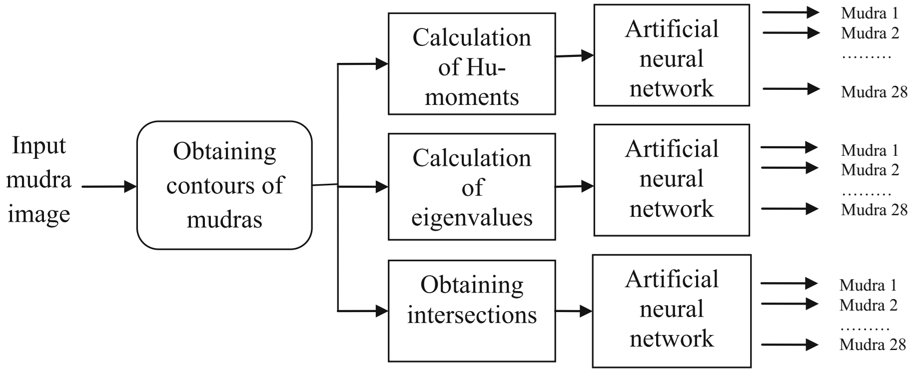


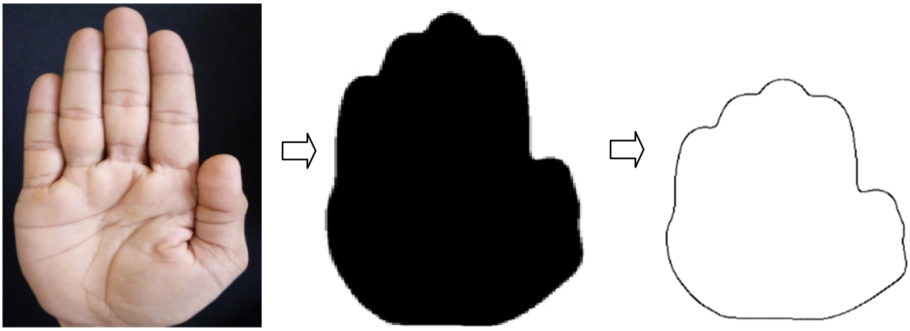
Fig. 5 Three stages of mudra classification methodology

3 Proposed Work

The dataset of Bharatanatyam mudra images is created by capturing the images of mudras. Natural lighting conditions are maintained for capturing the images of mudras. A 14.1 megapixels digital camera is kept at an appropriate fixed distance, 20 meters, from the performer. Bharatanatyam dancer is made to stand before a black background, any background colour contrast to the hand colour, and perform the mudras with right hand. The setup for capturing the mudra image is shown in Fig. 4. The proposed method is divided into three stages. In the first stage, acquired mudra images are preprocessed to extract contour of the mudra. In the second stage, Hu-moments, eigenvalues and intersections are extracted as features. In the third stage artificial neural network is used for classification of the unknown mudras. The deep learning technique, namely, convolutional neural network (CNN) is used for the classification of mudra dataset. The devised a three stage methodology is given in Fig. 5.

3.1 Obtaining Contours of Mudras

The work is made background independent as any background is converted to black background before further processing. The acquired mudra images are converted to binary images.



Pataka mudra   Binary form   Contour of mudra

Fig. 6 Obtaining contour from pataka mudra image

The canny edge detection technique is used to obtain boundary of the mudra. Bounding box is put to extract only the contour of the mudra. The extracted contour is resized to 200 × 200 pixels, for processing reasons. Figure 6 shows the process of obtaining contour of the pataka mudra from its image.

3.2 Feature Extraction

In this stage, Hu-moments, eigenvalues and horizontal and vertical intersections are extracted as features and used for training and testing artificial neural network.

3.2.1 Hu-Moments Calculation

Hu-moments represent suitable shape features are invariant to scaling, translation and rotation. The central moments are invariant under translation and can be made scale invariant by dividing through a properly scaled zeroth central moment. The central normalized moments are used to obtain invariants with respect to translation, scale and rotation. The zero order moment reflects the target area, the first order moment reflects the target center of mass and second order moment reflects the length of principal, auxiliary axis and the orientation angle of principal axis. The higher order moments describe the details of the image such as the target of the distortion and kurtosis distribution. The seven invariant moments are suitable for the description of overall shape of the target. Hence, we have used moments as features in our work.

The contour obtained is superimposed with a grid of size 8x8, which contains 64 cells, each of size 25 × 25 pixels. The cells in the grid are numbered from top left corner to bottom right corner and the contour is superimposed with grid as shown in Fig. 7. From each cell, the Hu-moments of the image are obtained and used as features. Opencv's 'HuMoments' function is used to obtain seven Hu-moments. The function takes seven 'central normalized moments' as input and produces seven Hu-moments as output.

Equations (1) through (13) give the calculation of Hu-moments.

Let f(x, y) be a point on a single channel image of size M x N, where, $x = 0, 1, \ldots M-1$; $y = 0, 1, \ldots N-1$. Then, ten, two dimensional spatial moments m_{ji} of order $(j+i)$ are computed as:

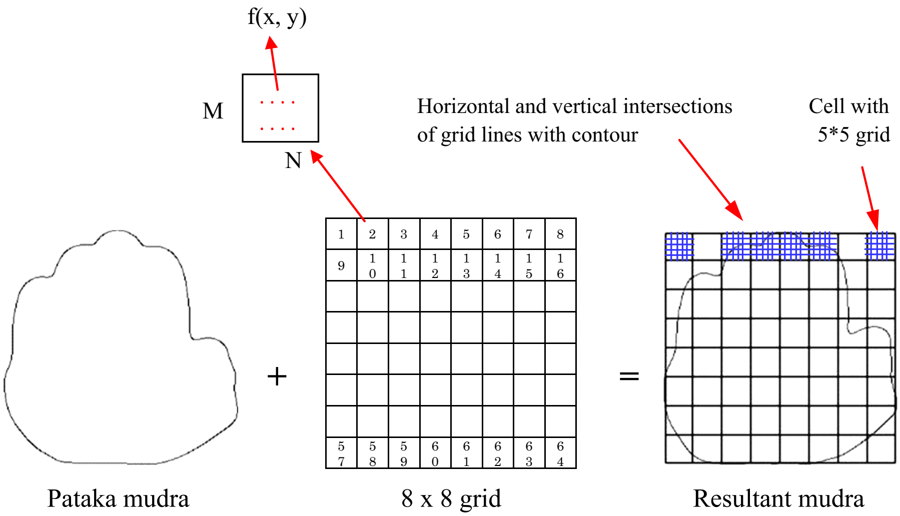


Fig. 7 Grid superimposed on contour

$$m_{ji} = \sum_{x,y} \left(f(x, y).x^j.y^i \right) \tag{1}$$

where i, j = 0, 1, 2, 3,…N − 1

The seven central moments mu_{ji} are computed as:

$$mu_{ji} = \sum_{x,y} \left(f(x, y).(x - x')^j.(y - y')^i \right) \tag{2}$$

where (x’,y’) is center of the mass, which is computed as:

$$x' = \frac{m_{10}}{m_{00}} \quad y' = \frac{m_{01}}{m_{00}} \tag{3}$$

The normalized central moments nu_{ji} are computed as:

$$nu_{ji} = \frac{mu_{ji}}{m_{00}^{(i+j)/2+1}} \tag{4}$$

$$mu_{00} = m_{00}, nu_{00} = 1 \tag{5}$$

$$nu_{10} = mu_{10} = mu_{01} = mu_{10} = 0 \tag{6}$$

Seven Hu-moments are calculated as:

$$hu[0] = \eta_{20} + \eta_{02} \tag{7}$$

$$hu[1] = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{8}$$

$$hu[2] = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{9}$$

$$hu[3] = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{10}$$

$$\begin{aligned} hu[4] = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{aligned} \tag{11}$$

$$hu[5] = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{12}$$

$$\mathrm{hu}[6] = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \tag{13}$$

where, $\eta_{ji} = nu_{ji}$.

The seven Hu-moments are added to get a single value as given in Eq. (14).

$$M = \mathrm{hu}[0] + \mathrm{hu}[1] + \mathrm{hu}[2] + \mathrm{hu}[3] + \mathrm{hu}[4] + \mathrm{hu}[5] + \mathrm{hu}[6] \tag{14}$$

The moment values of the cells are zeros, if the contour is not passing through them. Table 2 and Table 3 give the average values of cell-wise moments obtained for pataka and trishula mudras, respectively.

3.2.2 Fixing the Size of the Grid

The contours of the mudras are resized to a fixed size of 200 * 200 pixels. Consider a grid size of 2 * 2, having four cells, each of size 100 * 100 pixels. Figure 8a shows the scenario when 'tripataka' mudra is superimposed with the grid and Fig. 8b shows the scenario when 'shukatunda' mudra is superimposed with the grid. The mudra portion in cells 1, 3, and 4 is same in both the scenarios. The mudra portion is almost similar in the cell number 2, which produces least difference among the two mudras. Figure 8c and d show the scenario when the two mudras are superimposed with a grid of size 4 * 4, having 16 cells, each of size 50 * 50. It is observed from the figures that the cell number 4 has discriminating portion of mudra. To differentiate among the mudras which are almost looking similar to each other, the grid size is increased to 8 * 8, having 64 cells, each of size 25 * 25. Figure 8e and f show the scenario when the two mudras are superimposed with 8 * 8 grid. It is observed from the figures that cells 6, 15, and 22 have significantly discriminating portion of mudras. Hence, the grid of size 8 * 8 helps to produce most discriminating features among similar looking mudras. With this experimentation, we have decided the size of the grid to be 8 * 8.

3.2.3 Eigenvalues Calculation

Consider A ($n \times n$) matrix. If it occurs that v and w are scalar multiples as given in Eq. (15), eigenvalue equation of matrix A, then v is an eigenvector of the linear transformation of A and the scale factor λ is the eigenvalue corresponding to that eigenvector.

$$Av = w = \lambda v \tag{15}$$

Equation (15) is equivalently written as given in Eq. (16).

$$(A - \lambda I)v = 0 \tag{16}$$

where I ($n \times n$) is an identity matrix. Equation (16) has a non-zero solution v if and only if the determinant of the matrix ($A - \lambda I$) is zero. Therefore, the eigenvalues of A are values of λ that satisfy the Eq. (17).

$$|A - \lambda I| = 0 \tag{17}$$

Equation (17) is called the characteristic equation or the secular equation of A. The fundamental theorem of algebra implies that the characteristic polynomial of an n by n matrix A, being a polynomial of degree n, can be factored into the product of n linear terms as given in Eq. (18).

$$|A - \lambda I| = (\lambda_1 - \lambda)(\lambda_2 - \lambda)\ldots(\lambda_n - \lambda) \tag{18}$$

Table 2 Cell-wise average values of moments for pataka mudra

	1	2	3	4	5	6	7	8	
1	0	0.00197	0.01024	0.01084	0.01032	0.00842	0	0	8
9	0.00459	0.00197	0.01071	0	0	0.00679	0.00523	0	16
17	0.01067	0	0	0	0	0	0.01109	0.00171	24
25	0.01028	0	0	0	0	0	0.00925	0.00982	32
33	0.01086	0	0	0	0	0	0	0.01025	40
41	0.01103	0	0	0	0	0	0	0.01018	48
49	0.01177	0	0	0	0	0	0.00162	0.00968	56
57	0.00532	0.01046	0.01079	0.01106	0.01076	0.01087	0.00975	0.00113	64
	57	58	59	60	61	62	63	64	

Table 3 Cell-wise average values of moments for trishula mudra

	1	2	3	4	5	6	7	8	
1	0.02114	0.00551	0.00378	0.01167	0.01068	0.00089	0.00523	0.01085	8
9	0.01016	0.00852	0.0094	0.00798	0.00892	0.30994	0.01027	0.01076	16
17	0.00961	0.00493	0.01042	0.00782	0.01045	0.01024	0.00669	0.01105	24
25	0.01024	0.0044	0.00431	0.00429	0.00418	0.00519	0.00497	0.00616	32
33	0.00808	0.00576	0	0	0	0	0.0082	0.00348	40
41	0.00799	0.00412	0	0	0	0	0.00777	0.00325	48
49	0.00478	0.00651	0.00027	0	0	0	0.00975	0.00438	56
57	0.00039	0.0057	0.01019	0.01016	0.01005	0.01035	0.00747	0.00192	64
	57	58	59	60	61	62	63	64	

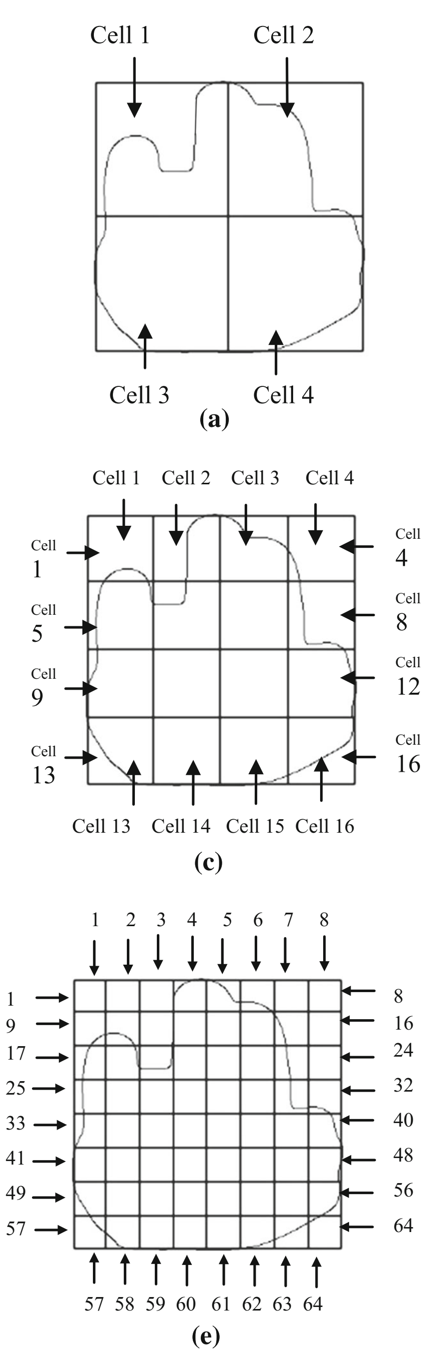


Fig. 8 Process of fixing the size of the grid: **a** Tripataka **b** Shukatunda **c** Tripataka **d** Shukatunda **e** Tripataka **f** Shukatunda

Each λ_i may be real but in general is a complex number. The numbers λ_1, λ_2, ... λ_n, which may not all have distinct values, are roots of the polynomial and are the eigenvalues of A. Hence, for an n by n matrix we obtain n eigenvalues. Eigen values are compact form of image features and represent principal components of an image matrix. They are widely used shape descriptors and hence we have used eigenvalues as features in our work.

The contour obtained is superimposed with a grid of size 8 * 8 as shown in Fig. 7. From each cell, the eigenvalues of the image are obtained and used as features. The image lying under the cell is extracted and its matrix form is created by using each pixel value of the image, which results in a square matrix of size 25 rows and 25 columns. Opencv's 'eigen' function is used to obtain eigenvalues. The function takes square matrix as an input and produces an array of eigenvalues as output, which contains the number of eigenvalues equal to the order of the square matrix. As the image under each cell is of size 25 * 25, we get 25 eigenvalues. We have considered largest eigenvalue as the feature. The eigenvalues of the cells are zeros, if the contour is not passing through them. Table 4 and Table 5 give cell-wise averages of eigenvalues obtained for pataka and trishula mudras, respectively.

3.2.4 Obtaining Intersections

The contour obtained is superimposed with a grid of size 8 * 8 as shown in Fig. 7. Each cell of the grid is further divided into a 5 * 5 grid, making the distance between two grid lines as 5 pixels. This is done on trial and error basis. The number of horizontal and vertical intersections of the grid lines with contour of the given mudra are counted and used as features. From each cell, the number of horizontal and vertical intersections is obtained and used as feature in the classification. Figure 9a and b show the average number of cell-wise horizontal intersections and vertical intersections obtained for pataka mudra and trishula mudra, respectively.

3.2.5 Computational Time Estimate

Feature extraction is an important stage of the proposed methodology. The computational efficacy of the proposed method is analyzed in terms of time and size of the dataset. We have used 100 images of each mudra resulting into a dataset of 2800 images. The time taken for extracting features from 100 images of each mudra is estimated. The time estimate is the sum of time taken for preprocessing, extracting features and storage of features. For example, the time taken for extracting intersections, moments and eigenvalues from 100 images of pataka mudra is estimated to be 69.48 s, 90.03 s and 82.21 s, respectively. The time taken for extracting each type of features from the images is given in Table 6.

3.3 Classification

The methodology is tested using artificial neural network for the considered three features, namely, Hu-moments, eigenvalues and intersections. The neural network tool of MATLAB 2015 is used for training and testing the neural network. The input layer consists of 64 neurons as we have 64 features from each image. The output layer consists of 28, 18 and 10 neurons corresponding to the number of classes corresponding to all the mudras, only non-conflicting mudras and only conflicting mudras, respectively. The feature dataset is appropriately divided and experiments are conducted by training and testing the artificial neural network. Table 7 gives the parameters used for setting up of the neural network.

Table 4 Cell-wise averages of eigenvalues for pataka mudra

	1	2	3	4	5	6	7	8	
1	0	1.695724	4.835981	4.91371	6.057128	1.72861	0	0	8
9	3.128539	4.571891	1.559714	0	0	3.992017	0.769712	0	16
17	4.41827	0	0	0	0	0	3.621492	0.024142	24
25	4.535091	0	0	0	0	0	3.91827	4.989862	32
33	3.7898	0	0	0	0	0	0	7.108514	40
41	1.754419	0	0	0	0	0	0	6.490003	48
49	4.392922	0	0	0	0	0	1.650069	4.181833	56
57	4.028153	2.685021	2.150764	2.528687	3.191872	4.07447	3.965937	1.514915	64
	57	58	59	60	61	62	63	64	

Table 5 Cell-wise averages of eigenvalues for trishula mudra

	1	2	3	4	5	6	7	8	
1	3.252012	1.20183	1.754118	4.846009	4.940444	3.76491	3.76491	6.007052	8
9	3.174819	4.347159	2.798489	2.997583	5.419755	1.874025	3.977814	5.057759	16
17	4.874012	1.913357	5.201824	2.97855	5.238841	4.2041	1.678634	4.238693	24
25	4.591669	1.37098	3.324568	2.997954	3.637363	3.436574	2.996696	3.171491	32
33	3.660431	1.632712	0	0	0	0	3.89958	0.893104	40
41	4.264128	0.779426	0	0	0	0	4.234566	1.287271	48
49	2.851502	1.815787	1.815787	0	0	0	3.83312	1.50982	56
57	0.485441	3.947352	3.022668	2.431182	3.046116	3.690503	3.742044	0.86617	64
	57	58	59	60	61	62	63	64	

0	1	3	2	3	1	0	0
1	4	0	0	0	3	2	0
5	1	0	0	0	2	3	0
5	0	0	0	0	1	3	3
5	0	0	0	0	1	1	5
5	0	0	0	0	0	0	5
5	0	0	0	0	0	1	4
2	3	0	0	1	2	3	1

Horizontal intersections

0	2	5	5	5	4	0	0
2	4	0	0	0	1	1	0
2	0	0	0	0	0	1	0
0	0	0	0	0	0	3	4
1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
3	0	0	0	0	0	1	4
0	5	5	5	5	5	5	1

Vertical intersections

(a)

1	1	1	4	4	1	2	3
5	4	3	3	4	1	5	5
4	2	5	4	4	5	1	5
3	2	3	2	2	2	2	3
4	1	0	0	0	0	4	1
4	1	0	0	0	0	4	1
2	3	1	0	0	0	4	1
1	3	2	0	1	2	2	1

Horizontal intersections

3	3	1	4	4	1	2	5
2	2	1	1	1	1	3	1
2	1	2	1	1	3	1	2
1	2	1	1	1	2	1	2
2	1	0	0	0	0	1	1
1	1	0	0	0	0	1	1
1	2	1	0	0	0	1	1
1	1	5	5	5	5	4	1

Vertical intersections

(b)

Fig. 9 Average number of intersections of mudras **a** Pataka mudra **b** Trishula mudra

3.3.1 Hu-Moments Based Classification

The artificial neural network is trained and tested by considering all 28 classes of mudras together. Overall classification accuracy is shown in Fig. 10. The overall average classification accuracy of single hand mudras is 97.1%. Classification accuracy of shukatunda mudra is 92.4% and tripataka mudra is 93.9%, as observed from Fig. 10. Minimum classification accuracy is 83.3% and maximum classification accuracy is 100%. It is observed that some of the mudras conflict with other mudras. Table 8 and Table 9 give list of non-conflicting mudras and conflicting mudras, respectively. Further, experiments are conducted separately for non-conflicting mudras and conflicting mudras. The average accuracy of non-conflicting single hand mudras is 99.5%. Minimum classification accuracy is 97% and maximum classification accuracy is 100%. The overall classification accuracy for non-conflicting mudras is shown in Fig. 11.

The average accuracy of conflicting single hand mudras is 96.03%. Minimum classification accuracy is 93.4% and maximum classification accuracy is 98.5%. The overall classification accuracy for conflicting mudras is shown in Fig. 12. The classification accuracy obtained for shukatunda mudra is 98.5%, as shown in Fig. 11 and 97.2% for tripataka mudra, as shown in Fig. 12. The increase in accuracy is attributed to experimentation on two homogeneous groups namely non-conflicting and conflicting, shown in Figs. 11 and 12, respectively.

3.3.2 Eigenvalues Based Classification

The artificial neural network is trained and tested by considering all 28 classes of mudras together. Overall classification accuracy is shown in Fig. 13. The overall average accuracy of

Table 6 Time taken for extracting features

Sl. no.	Mudra	Intersections	Moments	Eigenvalues
1	Pataka	69.48	90.03	82.21
2	Tripataka	75.03	86.57	86.28
3	Ardhapataka	65.17	78.52	79.46
4	Kartarimukha	70.46	98.98	100.80
5	Mayuraa	70.39	82.55	107.24
6	Ardhachandra	75.27	84.58	76.59
7	Arala	72.92	95.69	87.34
8	Shukatunda	74.14	87.38	86.48
9	Mushti	79.64	81.9	81.38
10	Shikhara	72.28	87.79	83.67
11	Kapitta	72.21	84.34	96.23
12	Katakamukha	72.89	81.53	89.77
13	Suchi	74.60	87.75	90.04
14	Chandrakala	70.17	82.79	82.96
15	Padmakosha	68.45	81.41	86.08
16	Sarpashirsha	74.50	88.09	87.25
17	Mrugashirsha	75.58	86.09	90.49
18	Simhamukha	76.16	86.84	88.95
19	Kangula	72.40	83.57	85.18
20	Alapadma	79.18	84.92	91.35
21	Chatura	87.53	84.98	86.73
22	Bhramara	73.49	84.17	90.34
23	Hamsassya	71.46	82.13	87.42
24	Hamsapakshika	77.06	84.24	86.83
25	Samdamsha	69.26	78.82	78.45
26	Mukula	75.48	80.90	81.43
27	Tamrachuda	75.57	86.11	89.07
28	Trishula	76.46	92.02	92.82

Table 7 Parameters used in implementation of neural network

Parameter	Value
Number of input neurons	64
Number of hidden neurons	128
Number of output neurons	28/18/10
Transfer function	Log sigmoid (LOGSIG)
Network type	Feed-forward backpropagation
Training function	TRAINGDX
Adaptation learning function	LEARNGDM
Performance metric	Mean Square Error (MSE)
Number of hidden layers	1

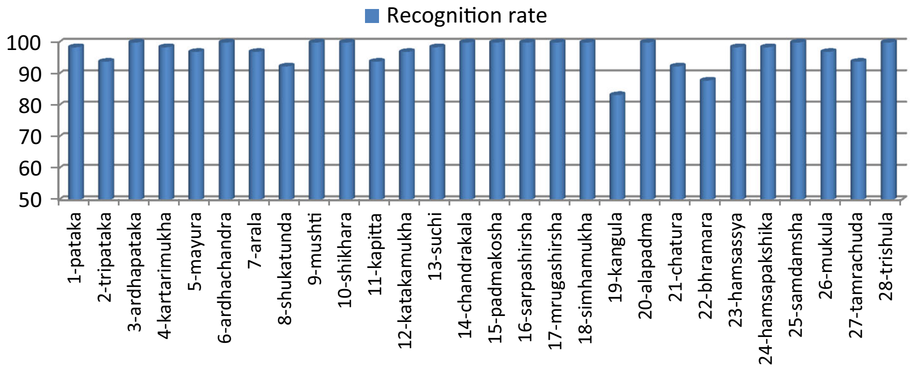


Fig. 10 Overall classification accuracy of asamyukta mudras

Table 8 Non-conflicting mudras

Mudra no.	Mudra	Mudra no.	Mudra	Mudra no.	Mudra
1	Pataka	8	Shukatunda	17	Mrugashirsha
3	Ardhapataka	9	Mushti	18	Simhamukha
4	Kartarimukha	10	Shikhara	20	Alapadma
5	Mayura	13	Suchi	25	Samdamsha
6	Ardhachandra	14	Chandrakala	27	Tamrachuda
7	Arala	15	Padmakosha	28	Trishula

Table 9 Conflicting mudras

Mudra no.	Mudra	Mudra no.	Mudra	Mudra no.	Mudra
2	Tripataka	19	Kangula	24	Hamsapakshika
11	Kapitta	21	Chatura	26	Mukula
12	Katakamukha	22	Bhramara		
16	Sarpashirsha	23	Hamsassya		

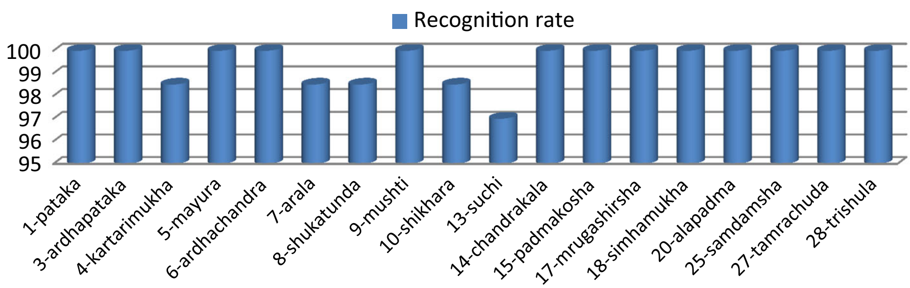


Fig. 11 Overall classification accuracy of non-conflicting mudras

single hand mudras is 98%. Classification accuracy of shukatunda mudra is 92.4% and tripataka mudra is 93.9%, as observed from Fig. 13. Minimum classification accuracy is 92.4%

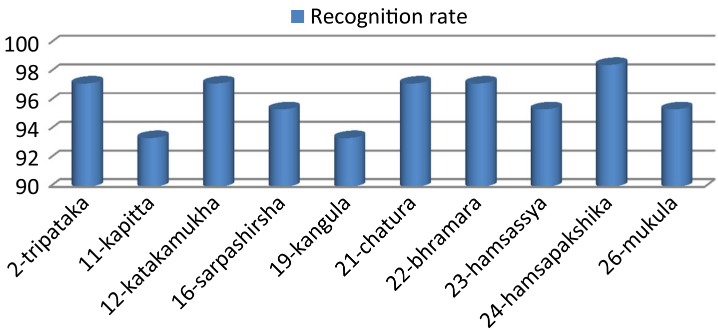


Fig. 12 Overall classification rate of conflicting mudras

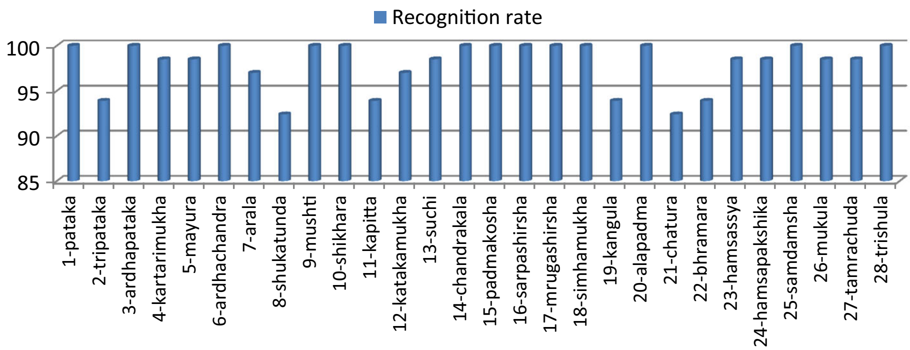


Fig. 13 Overall classification accuracy of asamyukta mudras

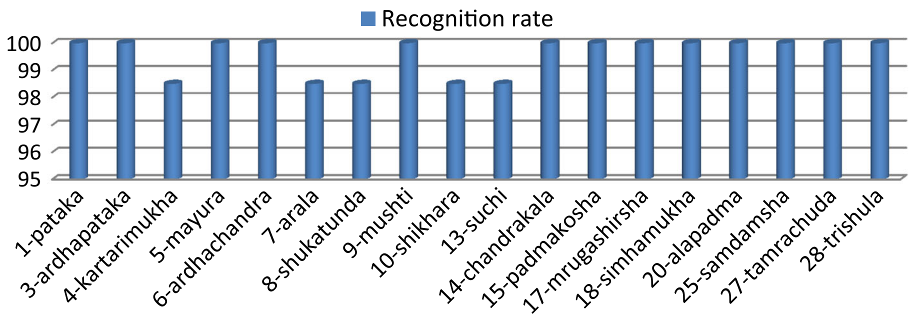


Fig. 14 Overall classification accuracy of non-conflicting mudras

and maximum classification accuracy is 100%. Further, experiments are conducted separately for non-conflicting mudras and conflicting mudras. The average accuracy of non-conflicting single hand mudras is 99.58%. Minimum classification accuracy is 98.5% and maximum classification accuracy is 100%. The overall classification accuracy for non-conflicting mudras is shown in Fig. 14.

The average accuracy of conflicting single hand mudras is 94.1%. Minimum classification accuracy is 92.4% and maximum classification accuracy is 97.4%. The overall classification accuracy for conflicting mudras is shown in Fig. 15. The classification accuracy obtained for shukatunda mudra is 98.5%, as shown in Fig. 14 and 97.4% for tripataka mudra, as shown

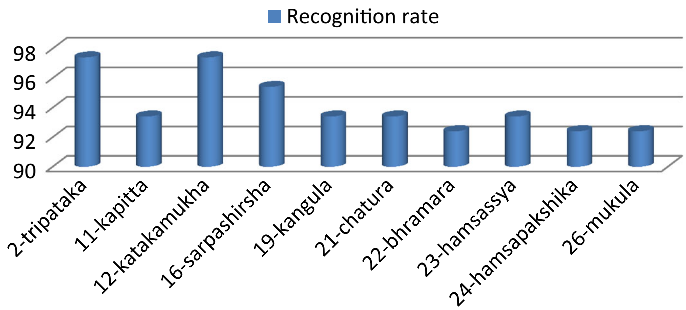


Fig. 15 Overall classification accuracy of conflicting mudras

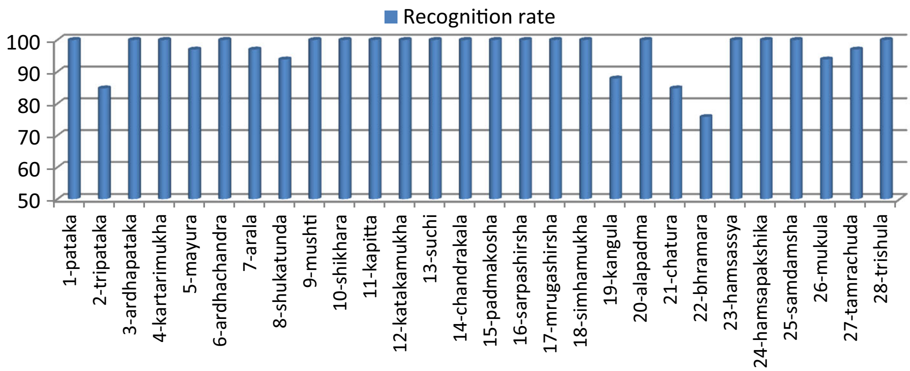


Fig. 16 Overall classification accuracy of asamyukta mudras

in Fig. 15. The increase in accuracy is attributed to experimentation on two homogeneous groups namely non-conflicting and conflicting, shown in Figs. 14 and 15, respectively.

3.3.3 Intersections Based Classification

The artificial neural network is trained and tested by considering all 28 classes of mudras together. Overall classification accuracy is shown in Fig. 16. The overall average accuracy of single hand mudras is 96.9%. Classification accuracy of shukatunda mudra is 93.9% and tripataka mudra is 84.8%, as observed from Fig. 16. Minimum classification accuracy is 75.8% and maximum classification accuracy is 100%. Further, experiments are conducted separately for non-conflicting mudras and conflicting mudras. The average accuracy of non-conflicting single hand mudras is 99.1%. Minimum classification accuracy is 93.9% and maximum classification accuracy is 100%. The overall classification accuracy for non-conflicting mudras is shown in Fig. 17. The average accuracy of conflicting single hand mudras is 95.2%. Minimum classification accuracy is 89.4% and maximum classification accuracy is 97%. The overall classification accuracy for conflicting mudras is shown in Fig. 18. The classification accuracy obtained for shukatunda mudra is 100%, as shown in Fig. 17 and 97% for tripataka mudra, as shown in Fig. 18. The increase in accuracy is attributed to experimentation on two homogeneous groups namely non-conflicting and conflicting, shown in Figs. 17 and 18, respectively.

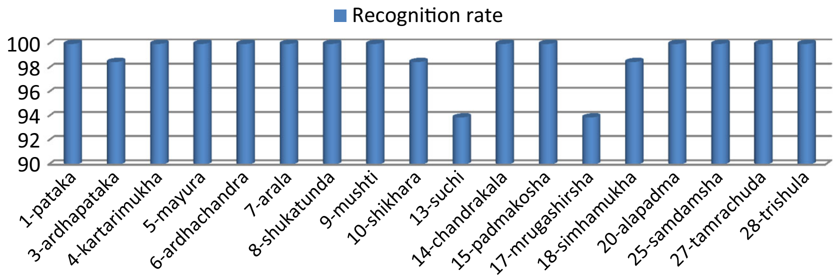
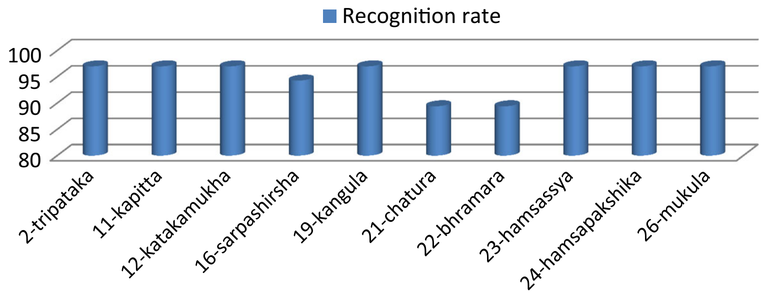


Fig. 17 Overall classification accuracy of non-conflicting mudras



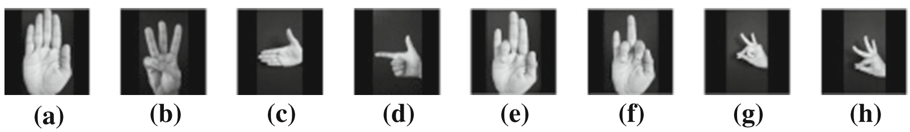


Fig. 19 Bharatanatyam dance mudra images used for deep learning: **a** Anjali **b** Trishula **c** Ardhachandra **d** Chandrakala **e** Tripataka **f** Shukatunda **g** Bhramara **h** Hamsassya

3.3.4 Deep learning Based Classification

Convolutional neural network (CNN) is used for the classification mudra images. The CNN takes raw images as input and learns the features from the images. The convolutional neural network is made up of input layer, convolution layer, rectified linear unit (ReLU) layer, max-pooling layer and fully connected layer or output layer.

The images in the dataset are converted to grayscale and are padded with boundaries of black pixels to make images exactly square. The grayscale images are resized to a size of 50×50 pixels. Figure 19 shows some of the images from the dataset. The convolutional layer consists of 20 filters of size 5×5. The convolutional layer is followed by a nonlinear activation function. We have used rectified linear unit (ReLU) activation function in this work. The convolutional layer is followed by max-pooling layer to reduce the number of parameters. These parameters are down-sampled with max-pooling of 2x2 regions. The fully connected layer combines all the features learnt by previous layers and classifies the images. The fully connected layer or output layer consists of number of nodes equal to the number of classes of mudras being classified. The architecture of proposed CNN is shown in Fig. 20.

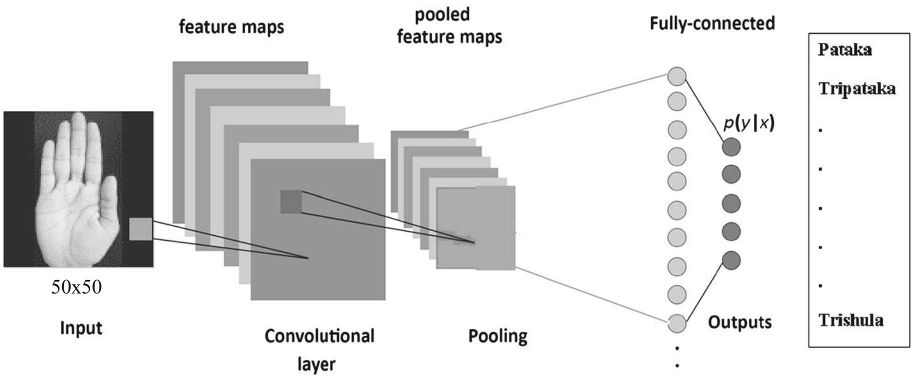


Fig. 20 Architecture of convolutional neural network for mudra images classification

Table 10 Experimental results of CNN on all 28 mudra classes

Sl. no.	Trained images from each class	Tested Images from each class	Epochs	Initial learning rate	Total time elapsed (s)	Accuracy obtained (%)
1	50	50	20	0.001	205.11	93.64
2	50	50	20	0.0001	199.34	92.21
3	50	50	25	0.001	260.45	93.78
4	50	50	25	0.0001	269.03	92.57
5	50	50	30	0.001	337.38	94.00
6	50	50	30	0.0001	301.31	92.57
7	75	25	20	0.001	319.28	94.71
8	75	25	20	0.0001	332.54	94.42
9	75	25	25	0.001	411.08	94.57
10	75	25	25	0.0001	401.78	94.42
11	75	25	30	0.001	477.13	94.57
12	75	25	30	0.0001	487.02	94.57

The dataset is randomly divided into training and testing set, wherein the number of images used for training and testing for each class are specified. We have set options for stochastic gradient descent with momentum for training. The training process is carried out by varying the number of images being used for training, number of epochs and initial learning rate. The trained CNN is tested on test dataset, which consists of the images from all that are not used for training the CNN. Table 10 gives the results of different experiments conducted on all 28 classes of mudras together. In an experiment, 75 images from each class are used for training and remaining 25 images from each class are used for testing. The initial learning rate is set to 0.001. The epochs are set to 20. We have obtained an accuracy of 94.71%. The total time elapsed is 319.28 s, which includes training time and also testing time.

Table 11 gives the results of different experiments conducted on non-conflicting mudra classes. In an experiment, the epochs are set to 30 and rest of the parameters is kept constant. We have obtained an accuracy of 98.22% and total time elapsed is 290.86 s. Table 12 gives the results of various experiments conducted on conflicting mudra classes. The experiment

Table 11 Experimental results of CNN on non-conflicting mudra classes

Sl. no.	Trained images from each class	Tested Images from each class	Epochs	Initial learning rate	Total time elapsed (s)	Accuracy obtained (%)
1	50	50	20	0.001	140.46	96.22
2	50	50	20	0.0001	140.35	95.00
3	50	50	25	0.001	173.81	96.56
4	50	50	25	0.0001	172.44	95.56
5	50	50	30	0.001	219.17	96.56
6	50	50	30	0.0001	218.52	95.67
7	75	25	20	0.001	195.91	98.00
8	75	25	20	0.0001	196.28	97.33
9	75	25	25	0.001	248.39	98.00
10	75	25	25	0.0001	244.00	97.56
11	75	25	30	0.001	290.86	98.22
12	75	25	30	0.0001	299.04	97.78

Table 12 Experimental results of CNN on conflicting mudra classes

Sl. no.	Trained images from each class	Tested Images from each class	Epochs	Initial learning rate	Total time elapsed (s)	Accuracy obtained (%)
1	50	50	20	0.001	58.98	92.60
2	50	50	20	0.0001	57.97	91.60
3	50	50	25	0.001	72.74	92.60
4	50	50	25	0.0001	72.72	92.00
5	50	50	30	0.001	87.61	92.60
6	50	50	30	0.0001	87.09	92.20
7	75	25	20	0.001	98.46	96.40
8	75	25	20	0.0001	101.73	94.80
9	75	25	25	0.001	124.84	97.60
10	75	25	25	0.0001	135.91	94.80
11	75	25	30	0.001	162.11	98.00
12	75	25	30	0.0001	167.05	94.80

where parameters are kept constant and epochs are set to 30 has produced an accuracy of 98.00%. The total time elapsed is 162.11 s.

Figure 21 shows deep learning classification accuracies obtained for all 28 mudra classes, non-conflicting and conflicting mudra classes. It is learnt that the accuracy of CNN increases as the number of training images and the number of epochs increase. The total time elapsed also increases with increase in number of training images and number of epochs.

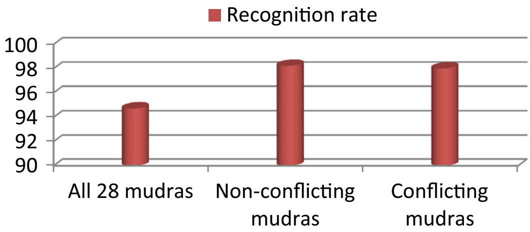


Fig. 21 Classification accuracies obtained using deep learning

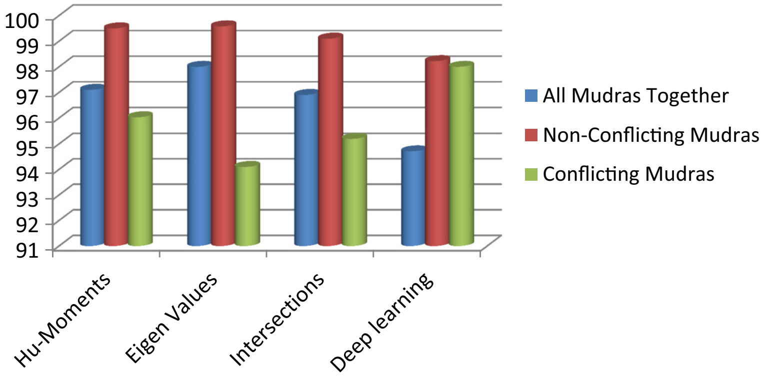


Fig. 22 Classification accuracies for three features and deep learning

4 Comparative Study

The methodology is tested for a total of 2800 images consisting of 100 images of each mudra. We have considered 15 different hands including male and female hands of varying sizes with different age groups. There are 8 female performers and 7 male performers. The performers are of age groups, namely, 8–15, 16–25, 26–35 and 35–50. Artificial neural network is trained and tested using Hu-moments, eigenvalues and horizontal and vertical intersections features. The experiments are also conducted on the dataset using convolutional neural network to corroborate the results with deep learning technique. Experiments are conducted by considering all classes of mudras together. It is observed that some of the mudras conflict with other mudras. Experiments are also conducted separately for non-conflicting mudras and conflicting mudras.

4.1 Classification Accuracies with Three Features and Deep Learning

The Hu-moments, eigenvalues and horizontal and vertical intersections features are compared in terms of their classification accuracies, when experimented with artificial neural network classifier. The convolutional neural network is trained and tested for classification of mudra images. The overall classification accuracy using Hu-moments is 97.1%, using eigenvalues is 98%, using intersections is 96.9% and using deep learning is 94.71%. Figure 22 shows the combined classification accuracies obtained by artificial neural network and convolutional neural network. Artificial neural network with eigenvalues has produced comparatively better classification accuracy when all mudras are considered together. Convolutional neural network has produced better results on group of conflicting mudras.

Table 13 Comparative study with related work

Sl. no.	Authors	Observations made	Remarks
1	Anami and Bhandage [1]	All 24 double hand mudras of Bharatanatyam dance are recognized. Mudras are discriminated as conflicting mudras and non-conflicting mudras. Vertical intersections, horizontal intersections and type of mudra are used as features. A Rule based classifier is developed. The work is implemented in any background colour, contrast to the hand colour	A rule based classifier is developed. Average accuracy across all 24 mudras is 95.25%
2	Sriparna et al. [15]	Center point for the boundary, eight spatial distances and fuzzy L-membership values are used as features. They have used black background	Recognition accuracy is 85.1%
3	Hariharan et al. [20]	Edge orientation histograms and shape based skeleton matching are used. They have used black background	System is tested for only 3 hands and 10 images of each hand
4	Kumar and Kishore [4]	A combination of single hand and double hand mudras, of Kuchipudi dance, are considered. Histogram of oriented gradients is used as feature and support vector machine is used as a classifier. They have used black background	An average accuracy of 90% is obtained
5	Solís et al. [5]	Normalized central moments are used as features. Multi-layer perceptron (MLP) is used as a classifier. Isolated MSL alphabets are recognized. They have used a uniform green background	An average accuracy of 93% is achieved
6	Adithya et al. [16]	Skin colour based segmentation is used. Shape based features are used and ANN is used as a classifier. They have used black background	Recognition accuracy is 91.11%
7	Dixit and Jalal [13]	Hu-moments and structural shape descriptors are used as features and multi-class support vector machine (MSVM) is used as a classifier. They have used black background	A recognition accuracy of 96% is obtained for a dataset of 720 images
8	Zadghorban and Nahvi [6]	Fourier descriptors, Hu-moments and Zernike moments, hand motion and dynamic time warping (DTW) are used as features. Hidden markov model and hybrid KNN-DTW algorithm are used for classification. They have used a uniform black background	Recognition accuracy is 93.73%
9	Fagiani et al. [10]	Hands centroid, Hu-moments, area, compactness, eccentricity and Hands orientation are considered as features. Hidden markov model (HMM) is used as a classifier. They have used a uniform black background	Overall recognition accuracy is 48.06%
10	Pradhan et al. [8]	Eigenvalues and eigenvectors are used as features. Euclidean distance is used to check similarity between known gesture and unknown gesture	A recognition accuracy of 65.38% is obtained for a dataset of 78 images
11	Sharma et al. [22]	Eigenvalues and eigenvectors are used as features. Artificial neural network is used as a classifier	Recognition accuracy is 98.5%

Table 13 continued

Sl. no.	Authors	Observations made	Remarks
12	Otiniano-Rodríguez et al. [19]	Public database of 2040 images stating 24 symbols classes is used. Two approaches are proposed for SLR using SVM as a classifier. They have used black background	Accuracy obtained by Hu-moments is 93% and Zernike moments is 96%
13	Singha and Das [17]	Eigenvalues and eigenvectors are used as features. Euclidean distance is used as a similarity measure to find unknown gesture	A recognition accuracy of 97% is obtained for a dataset of 240 images
14	Nguyen and Huynh [12]	Eigenvectors are used as features and Artificial neural network is used as a classifier	An overall accuracy of 94.3% is obtained for 24 gesture classes
15	Wang et al. [7]	The valley circle, Hu-moments, convexity, and compactness are used as features. They have used a uniform white background	10 different gesture classes are recognized
16	Fernando et al. [11]	Features such as hand contour, orientation histogram, convex hull, convexity defects and Hu-moments are considered. They have used a black background with small light adjustments	Recognition accuracy is 84%
17	Zaki and Shaheen [21]	Four components of sign languages such as hand shape, place of articulation, hand orientation, and movement are used. Kurtosis position, PCA and motion chain code are used as features and HMM as a classifier. They have used a black background	Overall recognition rate of 89.10% is achieved
18	Liu et al. [2]	p-Laplacian regularization is used with SVMs and KLSs for scene recognition. Experiments are conducted on Scene 67, Scene 15 and UC-Merced datasets	The performance is measured by using precision
19	Hong et al. [9]	Different features are combined using MDA and BP-NN is used to map 2D silhouettes to 3D poses. Experiments are conducted on Walking, HumanEva and Human3.6 M datasets	The recovery error has been reduced by 20%–25%
20	Yu et al. [3]	Multimodal features including visual and click features are used. A Deep-MDML method is developed. Experiments are conducted on LSIS and IRC datasets	The proposed rank model improves visual searching
21	Proposed work	All 28 single hand mudras of Bharatanatyam dance are recognized. Mudras are discriminated as conflicting mudras and non-conflicting mudras. Hu-moments, eigenvalues and horizontal and vertical intersections are used as features. Artificial neural network is used as a classifier. The convolutional neural network is trained and tested for classification of mudra images. The work is implemented in any background colour, contrast to the hand colour	Overall classification accuracy obtained with ANN by using Hu-moments, eigenvalues and intersections are 97.1, 98 and 96.9%, respectively. The CNN has produced an accuracy of 94.71%.

4.2 Comparative Study

The Table 13 gives a comparative study of the proposed method with the allied state-of-the-art works prevailing in the literature, as there are no similar works. The papers on sign language, hand gesture, moments, eigenvalue and deep learning based works are cited in the literature. This paper gives use of Hu-moments, eigenvalues and horizontal and vertical intersections as features along with artificial neural network classifier. In order to corroborate the results, a deep learning technique, convolutional neural network is used for classification of mudra images. The dataset, the number of hands, and also the number of mudras are large compared to the existing works carried out by other researchers. The overall average classification accuracies are acceptable.

5 Conclusion

The methodology is tested for all the 28 classes of single hand mudra images, containing 100 images of each mudra, giving a dataset of 2800 images, which is more than what earlier researchers have experimented. Further, earlier researchers have not discriminated the mudras as conflicting and non-conflicting. The methodology is tested with Hu-moments, eigenvalues and horizontal and vertical intersections features and artificial neural network classifier. A deep learning technique, convolutional neural network, is used to corroborate the results for the given image dataset. Overall classification accuracies using Hu-moments, eigenvalues, intersections and deep learning are 97.1%, 98%, 96.9% and 94.71%, respectively. The classification accuracies of considered features are acceptable. There is scope for improvement in overall classification accuracies of mudras in general and conflicting mudras in particular. It is also observed that the accuracy of deep learning classifier improves with increase in the number of training images and the number of epochs, which in turn takes more time for classification. Further, we wish to extend the methodology to double hand mudras. The developed methodology finds application in the area of fine arts and more particularly in effective e-learning of Bharatanatyam dance and online commentary during concerts.

Acknowledgements The authors wish to acknowledge Bharatanatyam dance teachers, namely, Mr. Gajanan V. and Ms. Mala T., Kala Kutir, Gadag, Karntaka, India and Mrs. Nandini S. Diwan, D/O. Pandit Badrinath Kulkarni, Prakash Nritya kala Mandir, Kolhapur,Maharstra, India for allowing us to capture images. Consent is also taken from the artist involved in the images.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Anami BS, Bhandage VA (2018) A vertical-horizontal-intersections feature based method for identification of bharatanatyam double hand mudra images. Multimed Tools Appl. https://doi.org/10.1007/s11042-018-6223-y
2. Liu W, Ma X, Zhou Y, Tao D, Cheng J (2018) p-Laplacian regularization for scene recognition. IEEE Trans Cybern 99:1–14. https://doi.org/10.1109/TCYB.2018.2833843
3. Yu J, Yang X, Gao F, Tao D (2017) Deep multimodal distance metric learning using click constraints for image ranking. IEEE Trans Cybern 47(12):4014–4024. https://doi.org/10.1109/TCYB.2016.2591583

4. Kumar KVV, Kishore PVV (2017) Indian classical dance mudra classification using HOG features and SVM classifier. Int J Electr Comput Eng (IJECE) 7(5):2537–2546. https://doi.org/10.11591/ijece.v7i1
5. Solís F, Martínez D, Espinoza O (2016) Automatic Mexican sign language recognition using normalized moments and artificial neural networks. Engineering 8:733–740. https://doi.org/10.4236/eng.2016.810066
6. Zadghorban M, Nahvi M (eds) (2016) An algorithm on sign words extraction and recognition of continuous Persian sign language based on motion and shape features of hands. In: Pattern analysis and applications. Springer, Berlin, pp 1–13
7. Wang M, Chen W-Y, Li XD (2016) Hand gesture recognition using valley circle feature and Hu's moments technique for robot movement control. Measurement 94:734–744
8. Pradhan A, Kumar S, Dhakal D, Pradhan B (2016) Implementation of PCA for recognition of hand gesture representing alphabets. Int J Adv Res Comput Sci Softw Eng 6(3):263–268
9. Hong C, Yu J, Wan J, Tao D, Wang M (2015) Multimodal deep autoencoder for human pose recovery. IEEE Trans Image Process 24(12):5659–5670. https://doi.org/10.1109/TIP.2015.2487860
10. Fagiani M, Principi E, Squartini S, Piazza F (2015) Signer independent isolated Italian sign recognition based on hidden Markov models. Pattern Anal Appl 18(2):385–402
11. Fernando M, Wijayanayake J (2015) Novel approach to use HU moments with image processing techniques for real time sign language communication. Int J Image Process (IJIP) 9(6):335
12. Nguyen T-N, Huynh H-H (2015) Static hand gesture recognition using principal component analysis combined with artificial neural network. J Autom Control Eng 3(1):40–45
13. Dixit K, Jalal AS (2013) Automatic Indian sign language recognition system. In: 2013 IEEE 3rd international advance computing conference (IACC). IEEE, pp 883–887
14. Premaratne P, Yang S, Zou Z, Vial P (2013) Australian sign language recognition using moment invariants. In: International conference on intelligent computing, LNAI 7996. Springer, Berlin, pp 509–514
15. Sriparna S, Ghosh L, Konar A, Janarthanan R (2013) Fuzzy L membership function based hand gesture recognition for Bharatanatyam dance. In: 2013 5th international conference on computational intelligence and communication networks (CICN). IEEE, pp 331–335
16. Adithya V, Vinod PR, Usha G (2013) Artificial neural network based method for Indian sign language recognition. In: Proceedings of 2013 IEEE conference on information and communication technologies (ICT 2013) Jeju Island, pp 1080–1085, April 2013
17. Singha J, Das K (2013) Indian sign language recognition using eigen value weighted euclidean distance based classification technique. Int J Adv Comput Sci Appl (IJACSA) 4(2):188–195
18. Liu Y, Yin Y, Zhang S (2012) Hand gesture recognition based on HU moments in interaction of virtual reality. In: 4th international conference on intelligent human-machine systems and cybernetics (IHMSC), vol 1. IEEE, pp 145–148
19. Otiniano-Rodríguez KC, Cámara-Chávez G, Menotti D (2012) Hu and Zernike moments for sign language recognition. In: Proceedings of the international conference on image processing, computer vision, and pattern recognition (IPCV), p 1
20. Hariharan D, Acharya T, Mitra S (2011) Recognizing hand gestures of a dancer. In: Kuznetsov SO, Mandal DP, Kundu MK, Pal SK (eds) Pattern recognition and machine intelligence. PReMI 2011. Lecture notes in computer science, vol 6744. Springer, Berlin
21. Zaki MM, Shaheen SI (2011) Sign language recognition using a combination of new vision based features. Pattern Recognit Lett 32(4):572–577. https://doi.org/10.1016/j.patrec.2010.11.013
22. Sharma P, Aneja A, Kumar A, Kumar S (2011) Face recognition using neural network and eigen values with distinct block processing. Int J Sci Eng Res 2(5). ISSN 2229-5518
23. Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, pp 2278–2324
24. Hu M-K (1962) Visual pattern recognition by moment invariants. IEEE Trans Inf Theory 8:179–187