The Scientific Temper (2023) Vol. 14 (4): 1374-1380
E-ISSN: 2231-6396, ISSN: 0976-8653
Doi: 10.58414/SCIENTIFICTEMPER.2023.14.4.46
https://scientifictemper.com/
RESEARCH ARTICLE

Integrated deep learning classification of Mudras of Bharatanatyam: A case of hand gesture recognition

Saba Naaz[1*], K.B. Shiva Kumar[2]

Abstract

Bharatanatyam is a famous Indian classical dance which incorporates the essence of hand gestures or mudras as a means of communication by performers. Bharatanatyam has around 28 single and 23 double hand gestures, respectively. Recognizing the gestures is challenging due to minor structural differences between the gestures and presence of gestures with higher structural similarity. This work proposes a deep learning strategy combining scale invariant feature transform features with convolutional neural network to address the challenge in accurate recognition of mudras. The gestures are segmented using active net based segmentation model to reduce the influence of background in gesture recognition. The gestures are then grouped based on similarity and convolutional neural network (CNN) is trained for each group to solve the problem of classification in presence of higher structural similarity. Convolutional neural network with structural conflict minimization kernel is used to classify the gestures. The proposed model attained an accuracy of 95% in classification of mudras and it has lower false positives of 2%.

Keywords: Activenet segmentation, Bharatanatyam, Convolutional neural network, Gesture recognition, Structural conflict minimization.

Introduction

Bharatanatyam is classical Indian dance with rigid code and conventions. It is based on the concept of coordinated face, foot hand, and body movements (Mallik *et al*., 2011). The most salient characteristics of Bharatanatyam are its Mudra. Mudra use hand gestures for non verbal communication between the performer and the audience. It can be used to visually convey outer events, inner feeling etc. Say Pathakam' mudra denotes joy, 'Mrigasirsham' mudra denote fear. Various meanings are indicated by different mudras. Mudras are of two types: Asamyukta and Samyukta (Mallick *et al*., 2021). In Bharatanatyam, there are 28 Asamyukta mudras, which are performed using a single hand, and 24 Samyukta mudras, which are performed using both hands. The structural differences between most of mudras are very less and often mudras are misinterpreted. Due to lack of availability of trained Bharatanatyam experts and increasing interest in self study of dance, many computer aided dance assistance tools are being developed (Jadhav & Mukundan, 2010). These tools assists dance practitioner to learn the art. Computer aided detection of mudras is a typical case of hand gesture recognition. Conventional methods for gesture recognition use hand crafted features like moments, shape descriptors etc along with machine learning (ML) techniques like artificial neural network (ANN), support vector machine (SVM) etc. Classification of mudras using conventional methods has higher false positives due to many structural conflicts in mudras (Anami & Bhandage, 2019). Among some of mudras shown in Figure 1, Chatura mudra conflicts with Sarpashirsha mudra, Hamsapakshika mudra conflicts with Katakamukha mudra, Hamsassya mudra conflicts with Bhramara mudra, and Mukula mudra conflicts with Kangula mudra.

This work proposes a deep learning (DL) strategy to mitigate the problem of misinterpretation of mudras due to structural conflicts. Scale invariant feature transform which is a conventional shape descriptor is combined with convolutional neural network (CNN) which is a DL technique for accurate recognitions of mudras. The main contributions are listed below:

- A novel segmentation algorithm based on active net and YOLOV3 is proposed to segment the mudras from

[1]Visvesvaraya Technological University, Belagavi, Karnataka, India.
[2]Dept of CSE, Don Bosco Institute Technology, Bengaluru, India
***Corresponding Author:** Saba Naaz, Visvesvaraya Technological University, Belagavi, Karnataka, India, E-Mail: sabazerdi@gmail.com

How to cite this article: Naaz, S., Kumar, K.B.S. (2023). Integrated deep learning classification of Mudras of Bharatanatyam: A case of hand gesture recognition. The Scientific Temper, **14**(4):1374-1380.
Doi: 10.58414/SCIENTIFICTEMPER.2023.14.4.46
Source of support: Nil
Conflict of interest: None.

Received: 12/10/2023 **Accepted:** 17/11/2023 **Published** : 27/12/2023

Figure 1: Conflicting mudras

background influences. This novel algorithm improvises the well known active net segmentation model with ML based active net link filtering using Naïve Bayesian algorithm.

- Integration of SIFT with CNN for resolving the mudra conflicts effectively. Mudra images are grouped based on structural similarity with SIFT features and CNN is trained for each group to discriminate structural similar mudras based on finer level differences.
- A novel structural conflict minimization kernel convolution operation at CNN to increase the accuracy of mudra classification.

The paper's remaining structure is outlined below: Section II provides a brief analysis of current approaches, categorized into mudra recognition and hand gesture recognition. Within mudra recognition, existing issues as well as research gaps are identified. Section III elaborates the proposed approach, integrating DL with traditional SIFT features for mudra recognition. The outcomes of the proposed method for various mudra classes are demonstrated and contrasted with prior studies in section IV. Finally, section V presents the conclusion and outlines the prospects for future research.

Related Work

Anami and Bhandage (2018a) extracted Eigen features from hand gestures. The features were classified using artificial neural network to mudras. The method was tested only for a limited set of mudras and the work did not address conflicting mudras. Saha *et al*. (2013) proposed a fuzzy membership approach for mudra recognition. Hand is segmented using texture segmentation approach. Eight spatial distances are calculated from the central point of boundary. Matching is based on this spatial distance vector using Fuzzy membership function. The method is not scale invariant. Parameshwaran *et al*. (2020) used transfer learning with CNN to improve the classification accuracy of mudras. To compensate for the lower data volume before applying CNN, data augmentation was applied. Two CNN models were used. Model 1 was pre-trained VGG16 architecture and model 2 was pre-trained Imagenet model. Even though the approach has the capability to obtain 98% accuracy, it was tested only for non conflicting scenarios. Anami and Bhandage (2018b) developed a three stage model for mudra recognition. Firstly, with the use of canny edge operator, gesture image's contour was attained. Cell features are extracted from grids drawn over the contour. Cell features are classified into 24 classes of mudras using rule based classifier. Kumar and Kishore (2018) extracted histogram of gradient (HoG) features and classified the features by employing SVM. The method wasn't scale invariant, also it was not tested against structural similarity between mudras. Kopuklu *et al*. (2019) modified the 3D CNN architecture for hand gesture recognition. The fully connected layer dimension is doubled and the training volume is increased for this dimension increase. There different data augmentation methods were used to increase the training volume. The model was tested for limited class of gestures and it suitability for 28 different mudras with higher structural similarities need to be explored. Sahoo *et al*. (2022) used pre-trained CNN model with score level fusion method for hand gestures recognition. Two pre-trained models of AlexNet as well as VGG-16 were utilised and score vectors of the two models are combined in weighted manner to provide the final score. Patil and Subbaraman (2019) extracted Hough transform based spatio temporal features and classified the features to gestures using ANN. Authors extracted fourier descriptor, geometrical feature and temporal feature and passed it as input to neural network to classify the gesture. The concepts used in this work for adapting to gestures of different scale can be considered even for mudra recognition. Fang *et al*. (2020) established a novel hand shape descriptor using Fisher vector and geometric features. By employing fisher vector, the geometric features of distances, angles and curvatures are extracted from hand gesture image and the features are encoded. Then, the encoded feature vector is classified using multi class support vector machine. Extraction of geometric features alone is not sufficient for mudra classification and this work cannot provide better performance for mudras. Gadekallu *et al*. (2022) fine tuned the hyper parameters of CNN to increase the hand gesture recognition performance. Hyper parameters of CNN are fine tuned using Harris Hawks Optimization algorithm. Hyper parameter optimization can be experimented for Mudra classification for higher performance. Lim *et al*. (2016) extracted feature covariance matrix from hand gesture image and used Eigen distance matching to recognize gestures. The method wasn't translation or scale invariant. Jain *et al*. (2021) suggested a DCNN based on Resnet to Indian classical dance categories. Resnet was taken as base model and 14 layers of Resnet were fine tuned to improve classification accuracy. But the method was not tested specifically for mudras. Kumar *et al*. (2017) proposed a multi-feature fusion approach for Indian dance classification. Features of ernike moments, Hu moments, shape signature, LBP features, and Haar are extracted in two stages of segmentation and post segmentation. The features are fused and passed as input to Adaboost multiclass classifier to recognize dance types. The method classifier dance types and does not address fine grained classification at level of hand gestures. Kishore *et al*. (2018) adapted CNN model for classifying mudras. Author integrated a stochastic pooling technique combining the advantages of min and max pooling technique. Though the approach is able to achieve 93% accuracy, it was not tested for conflicting similarities in mudras. Mozarkar and Warnekar (2013) retrieved salient features using quaternion fourier transform and classified the salient features to mudras using KNN classifier. But the features were not sufficient to detect finer level structural differences in mudras. Kishore *et al*. (2013) used shape and texture information to recognize Indian hand gestures. Segmentation of hand gesture is done using Chan-Vese (CV) active contour model. Shape and texture features are extracted and encoded to fisher vector. The fisher vector is passed as input to feed forward neural network for classifying the gestures. The method is very sensitive to scaling and translation of postures.

From the survey, most of the works on mudra recognition did not address the problem of self similarities in mudras and issue in accurate classification of mudras in presence of similarities. Most of the works attempted to improve the accuracy by using CNN hyper parameter tuning and data augmentation. But unless the structural differences are amplified at kernel level, the false recognition rate cannot be reduced.

Integrated Deep Learning Mudra Recognition

The proposed work brings an integration between tradition SIFT features and deep learning CNN to solve the problem of accurate recognition in presence of higher structural similarities across mudras. The proposed solution's architecture is depicted in Figure 2.

The proposed solution involves three components: two stage segmentation, SIFT based clustering of mudras,

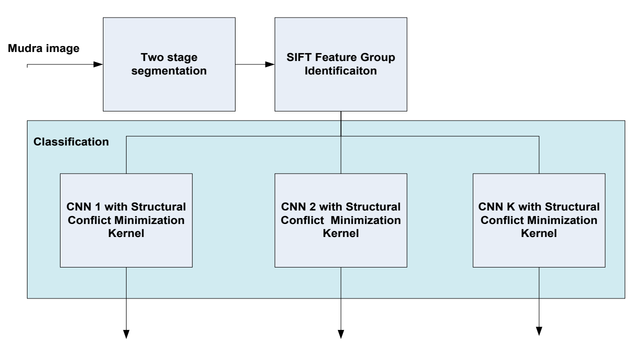

Figure 2: Architecture of proposed solution

Classification of mudras using CNN with novel structural conflict minimization kernel. Each of components are detailed in subsections (Figure 3).

Two stage segmentation

Effective segmentation of mudra is the first step in improving the recognition accuracy. This work proposes a two stage segmentation model eliminating the background influence to near zero level. YOLOv3 segmentation algorithm is used in first stage to detect the hand gesture region. For feature extraction, the YOLOv3 technique employs a CNN that integrates the Darknet-53 network design. The input image is separated into grids that are uniform in size. YOLOv3 detects the existence of objects inside every grid. By combining the nearby grids, the final bounding box around the object is formed. YOLOv3 becomes a novel feature by its incorporation of direct residual learning, which streamlines training and improves detection accuracy. After locating the hand gesture region in the image, it is cropped. The cropped image is then passed to next stage of active net segmentation to minimize the influence of backgrounds. A rectangular mesh is placed over the image. extended topological active net segmentation is employed on the provided image for accurate vehicle boundary representation.

During this process, the links within the mesh are grouped into different categories. The links exist at boundary need to be eliminated, thus, the remaining links denotes the object. To speed up the link removal procedure, a classification step is introduced. To enhance the efficiency of this classification process, a Naïve Bayesian (NB) classifier is constructed. For each link, a set of five features are extracted. These features are then fed into the trained NB classifier, which assigns one of the three predefined labels (as listed above) to each link based on its characteristics. The extracted features from the links are listed below:

- f1 – Link's Local minima
- f2 - Probable edge thickness
- f3 - LoGvalue around the starting node

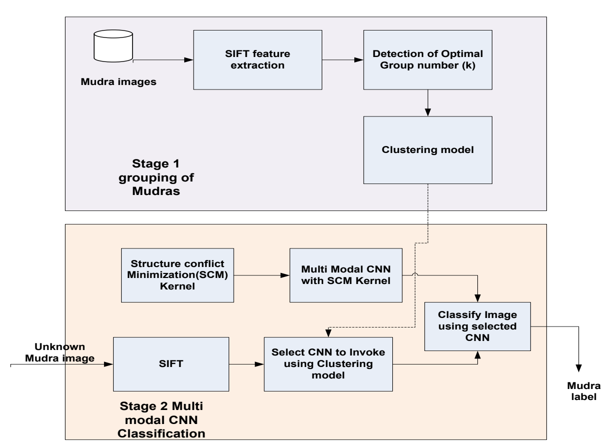

Figure 3: Classification process

- f4 - LoG value around the ending node
- f5 - Difference in dominant color around the two endpoint of the link

- *The f1(AB) is computed as follows*

Considering D as the midpoint of AB, the horizontal axis HH' is divided into equivalent points along the extension of the adjacent link. If the intensity progression along S_0, S_1, S_2 ... are consistently increasing, the discrepancy among the initial as well as final sampling points is considered as a potential feature for DH' direction. Similarly, if S_0, S_1, S_2 … are not consistently increasing, the feature value representing the DH' direction is considered as 0. Likewise, the features are determined for DH, DV, and DH', from which the maximum value is considered as a local minima of the link.

- *The f2 is computed as follows*

For every axis (DH', DH, DV, DV'), the maximal span of consecutively increasing or decreasing values among the sampled points is identified. The minimal value among these axis is considered as an indicator of the potential boundary's thickness.

Furthermore, the LoG is computed using a size of $5X5$ Gaussian filter around the initial node A and the terminal node B. It represents the existence of edges near the nodes A and B. It is calculated as

$$LoG(x,y) = -\frac{1}{\Pi\sigma^2}\left(1 - \frac{x^2+y^2}{2\sigma^2}\right)e^{\frac{x^2+y^2}{2\sigma^2}}$$

Using NB classifier, the links in the mesh are classified and every links with class label of "Links at the boundary" are eliminated (Figure 4).

SIFT based clustering of mudra

The mudra with higher structural similarity must be grouped and feature around structural differences must be focused for accurate classification of mudras. This work proposes grouping of mudra classes based on SIFT features. SIFT features are extracted from the segmented mudras. Top K SIFT points are extracted. The distance between the points are measured and divided the size of image to get the SIFT feature vector. The feature vectors are clustered using K mean clustering approach. The optimal amount of cluster is found using elbow method.

Elbow method is used for finding the k value. A graph is plotted by calculating the cost of clustering in terms of average distance between cluster points to its centroids for different values of k. As the k value increases, the cost drops. At certain k value, a elbow (similar to human hand) appears and this k value is selected as the optimal number of clusters.

An sample elbow point formation is shown in Figure 5.

In the Figure 5, the elbow point is formed at $k = 3$ and thus the optimal number of cluster is 3. Once the K groups and the mudras in each group is found, the training dataset is split into K groups.

Classification of mudras

This work uses CNN for recognition of mudras. Many works have proposed CNN hyper parameter optimization and data augmentation technique to improve the performance of CNN. In this work, a new structural conflict minimization kernel is used to improve the performance of CNN for mudra recognition (Figure 6).

CNN have the capability to learn more intricate features due to use of its convolutional kernels. But the structural similarity cannot be learnt in detail with default convolutional kernel. To address this issue, the present research introduces a CNN model with innovative convolutional kernel named structural conflict minimization (SCM) kernel. The SCM kernel is designed to enhance important areas for patterns, so that the features in those important areas become amplified in subsequent feature learning. As a result, this approach

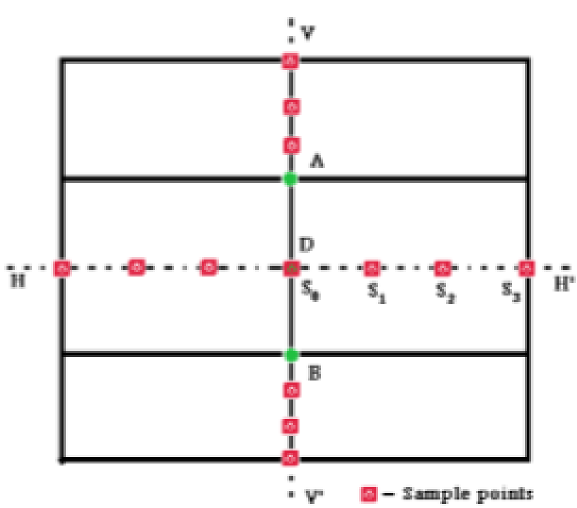

Figure 4: Link features

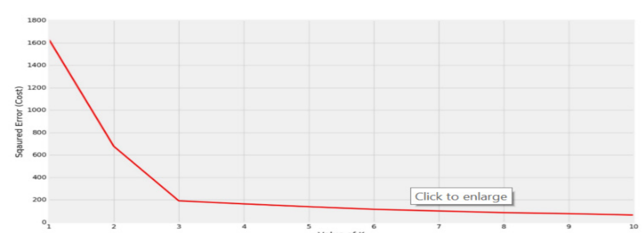

Figure 5: Elbow method

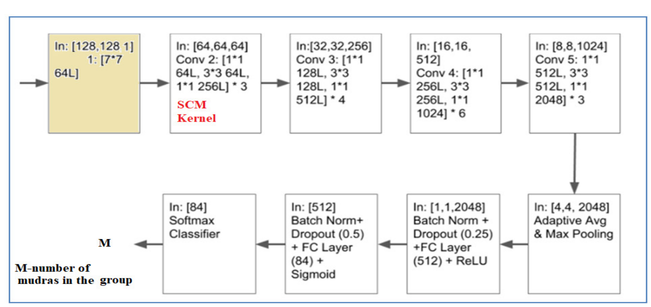

Figure 6: CNN architecture for mudra recognition

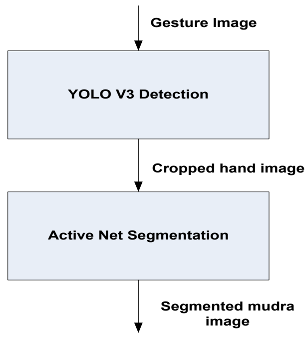

Figure 7: Two stage segmentation model

leads to an enhancement in classification accuracy. A binary mask is constructed for each group of mudras. This mask is constructed by finding the sectors of higher similarity in the mudra group and setting 1 for that sector and 0 for rest of the sectors. The SCM kernel convolution is applied by taking the image of size 64×64 and computing 8 local binary pattern LBP for it. The LBP results and binary masks are combined using logical AND operation. Subsequently, each of the 8 outcomes resulting from the AND operation is convolved with a 7×7 kernel and then summed up to generate the optimal output feature map

$$C(q) = \sum_{m=1}^{M}\sum_{n=1}^{N} AND(LBP(q), mask(m)).K(j)$$

Where,

M - number of times of masking,

N - number of LBP,

$mask(m)$ - binary mask applied to m^{th} LBP pattern.

The classification process is given in Figure 3 and two stage segmentation model is given in Figure 7.

For a input mudra image, SIFT features are extracted and SIFT feature vector is constructed based on relative distance between SIFT points as in section B. KNN classifier is used to find the group for the features and the corresponding CNN model is invoked with the segmented mudra image. The output of the CNN is the mudra class to which the image belongs.

Results

The effectiveness of the proposed approach is evaluated using the Indian classical mudras dataset (https://www.kaggle.com/datasets/ishanishah8/indian-classical-mudras-classification), which comprises 10 distinct mudra classes. For the remaining 18 mudra classes, images were sourced from various websites and the Chalearn dataset (https://www.kaggle.com/competitions/GestureChallenge/overview/data-description-2). The performance evaluation of the proposed solution is based on established metrics such as recall, precision, accuracy, and F1-score. Additionally, the performance evaluation of the two-stage segmentation method is conducted using structural similarity index (SSIM), spatial accuracy index (SAI), and Hausdorff distance.

SAI is calculated as,

$$SAI = \frac{C(R \cap T)}{C(R) + C(T)}$$

Where,

R - segmentation result,

T - ground truth,

$C(X)$ - cardinality of X.

SSIM is calculated as,

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_1)}$$

Where

μ_x	Mean
σ_x	Variance
σ_{xy}	Co variance
x	Segmented result
y	Ground truth

Hausdorff distance is the maximum distance between two contours

$$d_H(R,T) = max\begin{cases}\sup\inf\ d(r,t), r\varepsilon R^{t\varepsilon T} \\ \sup\inf\ d(r,t), t\varepsilon T^{r\varepsilon R}\end{cases}$$

It calculates the worst case of segmentation. Lower value of $d_H(R,T)$ represents higher segmentation performance.

The performance evaluation is conducted for individual classes of mudras. The effectiveness of the proposed solution is contrasted against Anami and Bhandage (2018b) vertical-horizontal-intersections feature-based method, Anami and Bhandage (2018a) eigenvalue-based matching technique, and Jain *et al.* (2021) DCNN approach. The average performance across all 28 mudra classes are computed and presented in Table 1.

The average accuracy in proposed solution is atleast 1.4% higher contrasted to Jain *et al*, 11.7% higher contrasted to Anami and Bhandage (2018b) and 18.6% higher contrasted to Anami and Bhandage (2018a). The accuracy has increased in proposed solution because of the division of mudra in groups based on SIFT features and dedicating classifier model for each group. Jain *et al.* used dense network but due to overloading the same CNN for classes, its accuracy was 1.4% lower compared to proposed solution. Anami and Bhandage (2018b) and Anami and Bhandage (2018a) used intersection and Eigen features which were not able to isolate mudras based on structural conflicts.

The box and whisker plot for accuracy, precision, recall is given in Figures 8, 9 and 10.

Table 1: Performance comparison

Methods	Precision	Accuracy	Recall	F1-score
Proposed	0.951	0.949	0.938	0.941
Jain *et al.* (2021)	0.937	0.935	0.936	0.931
Anami and Bhandage (2018b)	0.835	0.832	0.832	0.832
Anami and Bhandage (2018a)	0.787	0.763	0.764	0.759

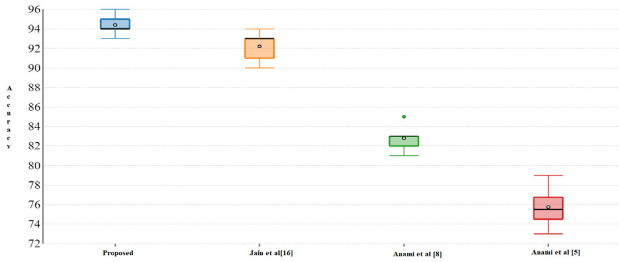

Figure 8: Box whisker plot for accuracy

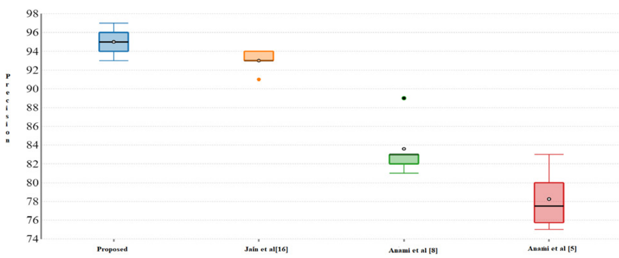

Figure 9: Box whisker plot for precision

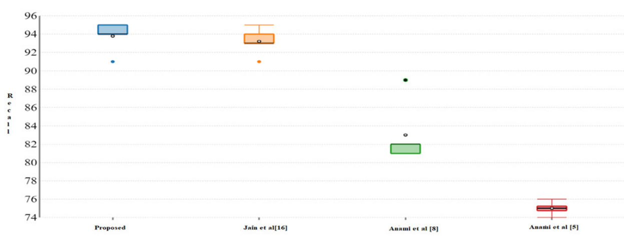

Figure 10: Box whisker plot for recall

The results for detection of mudras using YOLO V3 algorithm is measured and the result is given in Figure 7. Use of YOLO V3 is able to detect mudras with accuracy of about 93%. Figure 11 presents raining and validation accuracy for proposed model.

The performance for proposed two stage segmentation is computed for all 28 class of mudras and the segmentation results are compared to contour based segmentation proposed in Anami and Bhandage (2018b).

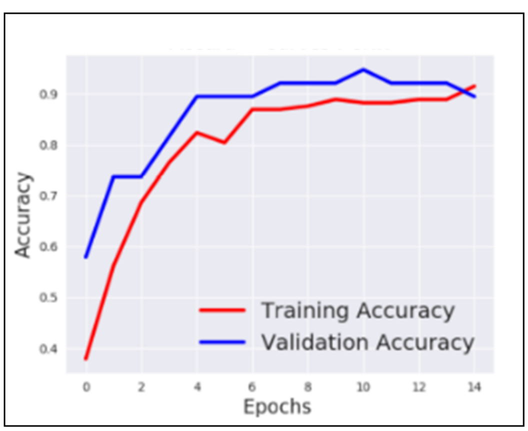

Figure 11: Training and validation accuracy for proposed model

Table 2: Performance comparison

Metrics	*Proposed*	*Anami and Bhandage (2018b)*
SAI	0.85	0.72
SSIM	0.88	0.81
Hausdorff distance	18	29

Table 3: Performance analysis of proposed SCM kernel with CNN-default kernel

Measures	Default kernel	SCM Kernel
Precision	0.922	0.951
Recall	0.921	0.938
Accuracy	0.910	0.949
F1-measure	0.921	0.941

Table 4: Performance analysis of proposed solution with and without grouping

Measures	*Without grouping*	*With grouping*
Precision	0.891	0.951
Recall	0.873	0.938
Accuracy	0.892	0.949
F1-measure	0.904	0.941

The segmentation results are given in Table 2.

The results indicate that the proposed solution achieves a higher SAI value contrasted to Anami *et al.*, highlighting the nearness of the segmentation result to the ground truth in the proposed solution. On the other hand, the SSIM value is lower in the proposed solution due to the segmentation boundaries closely aligning with the ground truth. Additionally, the proposed solution exhibits a lower Hausdorff distance, thus effectively demonstrating the segmentation's accuracy.

The performance of the CNN with default kernel and the proposed SCM kernel is measured and the result is given in Table 3.

The accuracy has increased by 3.9% due to use of SCM kernel compared to default kernel. This is due of consideration of structural differences around the SIFT keypoints with the SCM kernel.

The performance of the proposed solution with and with grouping of mudras is measured and the result is given in Table 4.

Due to grouping, the accuracy of mudra recognition has increased by 6%. Grouping allowed CNN to focus more on structural differences and this has helped to manage the structural conflicts in a better way in the proposed solution.

Conclusion

This work integrated deep learning CNN with traditional SIFT features to handled the structural conflict problem in mudra recognition. As part of the proposed work, a two stage segmentation method and novel structural conflict minimization based kernel is proposed. The proposed solution achieved an accuracy of 94.9% which is 1.4% higher than prior researches. The proposed two stage segmentation model is able to provide better isolation of background with structural similarity index at 0.88. This proposed solution was tested only for single hand mudras and it can be extended for double hand mudras as part of future work. The performance of CNN used in the model can be improved further by hyper-parameter optimization with fitness function designed to minimize errors in mudra classification.

References

Anami, B. S., & Bhandage, V. A. (2018a). Artificial neural network based identification of Bharatanatyam Mudra images using eigen values. *International Journal of Applied Pattern Recognition*, *5*(3), pp. 191-205. doi: 10.1504/IJAPR.2018.10016094.

Anami, B. S., & Bhandage, V. A. (2018b). A vertical-horizontal-intersections feature based method for identification of bharatanatyam double hand mudra images. *Multimedia Tools and Applications*, *77*(23), 31021-31040. https://doi.org/10.1007/s11042-018-6223-y

Anami, B. S., & Bhandage, V. A. (2019). A Comparative Study of Suitability of Certain Features in Classification of Bharatanatyam Mudra Images Using Artificial Neural Network. *Neural Processing Letters*, *50*(1), 741-769. https://doi.org/10.1007/s11063-018-9921-6

Fang, L., Liang, N., Kang, W., Wang, Z., & Feng, D. D. (2020). Real-time hand posture recognition using hand geometric features and fisher vector. *Signal Processing: Image Communication*, *82*, 115729. https://doi.org/10.1016/j.image.2019.115729

Gadekallu, T. R., Srivastava, G., Liyanage, M., Iyapparaja, M., Chowdhary, C. L., Koppu, S., & Maddikunta, P. K. R. (2022). Hand gesture recognition based on a Harris hawks optimized convolution neural network. *Computers and Electrical Engineering*, *100*, 107836. https://doi.org/10.1016/j.compeleceng.2022.107836

https://www.kaggle.com/competitions/GestureChallenge/overview/data-description-2

https://www.kaggle.com/datasets/ishanishah8/indian-classical-mudras-classification

Jadhav, S., & Mukundan, S. (2010). A Computational Model for Bharata Natyam Choreography. *International Journal of Computer Science and Information Security (IJCSIS)*, *8*(7), 231-233.

Jain, N., Bansal, V., Virmani, D., Gupta, V., Salas-Morera, L., & Garcia-Hernandez, L. (2021). An Enhanced Deep Convolutional Neural Network for Classifying Indian Classical Dance Forms. *Applied Sciences*, *11*(14), 6253. https://doi.org/10.3390/app11146253

Kishore, P. V. V., Kishore, S. R. C., & Prasad, M. V. D. (2013). Conglomeration of hand shapes and texture information for recognizing gestures of Indian sign language using feed forward neural networks. *International Journal of engineering and Technology (IJET)*, *5*(5), 3742-3756.

Kishore, P. V. V., Kumar, K. V. V., Kumar, E. K., Sastry, A. S. C. S., Kiran, M. T., Kumar, D. A., & Prasad, M. V. D. (2018). Indian classical dance action identification and classification with convolutional neural networks. *Advances in Multimedia*, *2018*, 5141402. https://doi.org/10.1155/2018/5141402

Köpüklü, O., Gunduz, A., Kose, N., & Rigoll, G. (2019, July 11). Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, Lille, France, 14-18 May 2019 (pp. 1-8). IEEE. doi: 10.1109/FG.2019.8756576

Kumar, K. V. V., & Kishore, P. V. V. (2018). Indian classical dance mudra classification using HOG features and SVM classifier. In Satapathy, S., Bhateja, V., Das, S. (Eds.), *Smart Computing and Informatics; Smart Innovation, Systems and Technologies, Smart Computing and Informatics: Proceedings of the First International Conference on SCI 2016*, (vol. 77, pp. 659-668). Singapore: Springer. https://doi.org/10.1007/978-981-10-5544-7_65

Kumar, K. V. V., Kishore, P. V. V., & Kumar, D. A. (2017). Indian classical dance classification with adaboost multiclass classifier on multifeature fusion. *Mathematical Problems in Engineering*, *2017*, 6204742. https://doi.org/10.1155/2017/6204742

Lim, K. M., Tan, A. W. C., & Tan, S. C. (2016). A feature covariance matrix with serial particle filter for isolated sign language recognition. *Expert Systems with Applications*, *54*, 208-218. https://doi.org/10.1016/j.eswa.2016.01.047

Mallick, T., Das, P. P., & Majumdar, A. K. (2021). Bharatanatyam dance transcription using multimedia ontology and machine learning. In Mukhopadhyay, J., Sreedevi, I., Chanda, B., Chaudhury, S., & Namboodiri, V. P. (Eds.), *Digital Techniques for Heritage Presentation and Preservation* (pp. 179-222). Cham: Springer. https://doi.org/10.1007/978-3-030-57907-4_10

Mallik, A., Chaudhury, S., & Ghosh, H. (2011). Nrityakosha: Preserving the intangible heritage of indian classical dance. *Journal on Computing and Cultural Heritage (JOCCH)*, *4*(3), 11. https://doi.org/10.1145/2069276.2069280

Mozarkar, S., & Warnekar, C. S. (2013). Recognizing bharatnatyam mudra using principles of gesture recognition gesture recognition, *International Journal of Computer Science and Network*, *2*(4), 46-52.

Parameshwaran, A. P., Desai, H. P., Sunderraman, R., & Weeks, M. (2020, April 09). Transfer Learning for Classifying Single Hand Gestures on Comprehensive Bharatanatyam Mudra Dataset. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, 16-17 June 2019 (pp. 508-510). IEEE. doi: 10.1109/CVPRW.2019.00074

Patil, A. R., & Subbaraman, S. (2019). A spatiotemporal approach for vision-based hand gesture recognition using Hough transform and neural network. *Signal, Image and Video Processing*, *13*(2), 413-421. https://doi.org/10.1007/s11760-018-1370-1

Saha, S., Ghosh, L., Konar, A., & Janarthanan, R. (2013, November 11). Fuzzy L Membership Function Based Hand Gesture Recognition for Bharatanatyam Dance. In *2013 5th International Conference and Computational Intelligence and Communication Networks*, Mathura, India, 27-29 September 2013 (pp. 331-335). IEEE. doi: 10.1109/CICN.2013.75

Sahoo, J. P., Prakash, A. J., Pławiak, P., & Samantray, S. (2022). Real-Time Hand Gesture Recognition Using Fine-Tuned Convolutional Neural Network. *Sensors*, *22*(3), 706. https://doi.org/10.3390/s22030706