

PAPER • OPEN ACCESS

Coot optimization based Enhanced Global Pyramid Network for 3D hand pose estimation

To cite this article: Pallavi Malavath and Nagaraju Devarakonda 2022 *Mach. Learn.: Sci. Technol.* **3** 045019

View the [article online](#) for updates and enhancements.

You may also like

- [Analysis of Crossover-Induced Capacity Fade in Redox Flow Batteries with Non-Selective Separators](#)

Venkat Pavan Nemani and Kyle C. Smith

- [The 2013 November 12 Solar Energetic Electron Event Associated with Solar Jets](#)

Wen Wang, , Andrea Francesco Battaglia et al.

- [Structural Design of TiO₂/Si Hybrid Photoelectrode and Pt-Free Counter Photoelectrodes for Charge Carrier Separation in Water-Splitting Reactions](#)

Vytautas Kavaliunas, Yoshinori Hatanaka, Yoichiro Neo et al.



PAPER

OPEN ACCESS

RECEIVED
7 June 2022REVISED
12 October 2022ACCEPTED FOR PUBLICATION
1 November 2022PUBLISHED
25 November 2022

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Coot optimization based Enhanced Global Pyramid Network for 3D hand pose estimation

Pallavi Malavath* and Nagaraju Devarakonda

School of computer science & Engineering, VIT-AP University, Amaravati, India

* Author to whom any correspondence should be addressed.

E-mail: pallavimalavath.phd@gmail.com**Keywords:** hand pose estimation, Enhanced Global Pyramid Network, DetNet, pose correction network, Coot optimization and sign language

Abstract

Due to its importance in various applications that need human-computer interaction (HCI), the field of 3D hand pose estimation (HPE) has recently got a lot of attention. The use of technological developments, such as deep learning networks has accelerated the development of reliable 3D HPE systems. Therefore, in this paper, a 3D HPE based on Enhanced Global Pyramid Network (EGPNet) is proposed. Initially, feature extraction is done by backbone model of DetNetwork with improved EGPNet. The EGPNet is enhanced by the Smish activation function. After the feature extraction, the HPE is performed based on 3D pose correction network. Additionally, to enhance the estimation performance, Coot optimization algorithm is used to optimize the error between estimated and ground truth hand pose. The effectiveness of the proposed method is experimented on Bharatanatyam, yoga, Kathakali and sign language datasets with different networks in terms of area under the curve, median end-point-error (EPE) and mean EPE. The Coot optimization is also compared with existing optimization algorithms.

1. Introduction

Most of the human activities in daily life are communicating with others. Navigation, manipulation and gesture are some of the basic interactions. Perhaps people are supported by ground to navigate complex situations and avoid obstacles [1]. They always use their hands and fingers in a large number of tasks to communicate with others (through communication gestures) or with the physical world around them. This recommends that movement and communication with the environment are strongly intertwined [2]. In virtual reality and human-computer interaction (HCI), 3D hand pose estimation (HPE) is used because it can provide basic details for interrelating with objects and making gestures. The HPE of single depth images has recently gained a lot of scientific interest due to the availability of advanced cameras. However, hand posture estimation remains a very challenging issue due to the high degree of severe self-obstruction, complexity of the hand accent, the noise and the holes in the deep picture, the large variation of views and the self-similarity of the fingers [3].

By the advances in image processing schemes, image processing applications are currently implemented with a computer-assisted system. Gesture recognition is an area that has seen great progress in this field [4]. It is well known that Indian classical dance (ICD) forms like Bharatanatyam, Kathakali and almost 16 types of Indian traditional dances use 51 types of gestures known as mudras. These signs are communicative and meaningful hand postures. Under appropriate circumstances, the gesture sensor must be able to recognize the gesture [5]. This mudra pose estimation progression is used in the proposed method so as to identify the Bharatanatyam, Kathakali, yoga and sign language mudras.

The traditional method of HPE is to use a deformable hand model to fit the deep figure [6], which is computationally complex but which allows the intermediate frame to be taken as the initial value of the next frame. However, the error will accumulate hence; multiple frames should be considered simultaneously rather than the same frame [7]. However, due to the diversity of appearance produced by posture changes

and motion, such as occlusion, illumination, facial texture and so on, estimating hand position directions is extremely challenging in practice. Several techniques have been presented over the years to address these issues. In this paper, 3D mudra estimation network is introduced. The key contribution of the proposed method is follows,

- In this paper, an Enhanced Global Pyramid Network (EGPNet) with 3D Pose correction network for HPE is proposed.
- Initially, the backbone model of DetNetwork (DetNet) is hybridized with EGPNet for extraction the features from the given dataset. After the feature extraction, the HPE is performed based on 3D Pose correction network.
- After estimating the hand mudra, the Coot optimization is used to enhance the estimation accuracy.
- Through this proposed network, the yoga, Kathakali Bharatanatyam mudras and sign languages are estimated from the corresponding datasets.
- The effectiveness of the proposed scheme is evaluated in terms of area under the curve (AUC), median end-point-error (EPE), and mean EPE through various networks and optimization algorithms.

The rest of the paper is organized as follows. The existing HPE approaches are explained in section 2. Section 3 discusses the proposed HPE process. Section 4 discusses the results of the experiments and finally, the work is concluded in section 5.

2. Related works

Due to the wide range of applications, the recent researches on gesture recognition and hand detection are attracted increasing interest in the applications of sign language recognition, virtual reality, hand-action analysis and driver hand behavior monitoring. Recently numerous, methods have been offered to developing a strong algorithm in complex environments. Then the pose estimation approximately seems solved for scenes with isolated hands. But still it is difficult to analyses cluttered prospects where hands may be relating with the nearest surfaces and objects.

Oberweger *et al* [8] introduced numerous convolutional neural network (CNN) models to predict the 3D joint positions of a hand. Initially, 3D pose has been introduced and next enhance the accuracy and prediction reliability. Both of these allow us to significantly surpass on many challenging criteria, both in accuracy and computational analysis. To enhance the performance of HPE, Chen *et al* [3] proposed a Pose guided Region Ensemble Network (Pose-REN). Pose-REN extracts regions from CNN feature maps based on the estimated pose to provide the best features for pose estimation. To degrade the advanced hand position, the derived feature regions are combined based on tree-structured fully connections. The final hand pose was achieved using an iterative cascaded system and a repeating layering scheme.

To capture complex hand structures, Ge *et al* [9] proposed a Hand PointNet that represents the visible surface of the hand for pose regression. The low-dimensional portrayal of the 3D hand posture is perfectly reversed as a result of this. A fingertip refinement network was designed to enhance the accuracy of the fingertips by directly inputting the nearby points of the estimated fingertip location and optimizing the location of the fingertips. Dance forms are difficult to comprehend since they are multimedia in nature and stretch over time and space. Capturing and analyzing the dance's multimedia elements can help preserve cultural heritage, creating video referral systems, and helping learners to use practice methods. Mallick *et al* [4] were tried to solve three basic problems of dance analysis in order to understand the concept of dance forms. Next, identify key poses using machine learning and deep learning approaches. Finally, the hidden Markov model was used to recognize the dance sequence. The multi-model data from Bharatanatyam was used to analyze and create an annotated data set for ICD research.

Human pose classification is the most challenging part of research in modern days. It is widely supported in understanding a person's posture and the sequence of its next actions. Several standard human pose databases have been developed and an extensive research is underway. Priya and Arulselvi [10] main goal is to create a multiview database with innovative actions that differ from the normal and taken from karate martial arts and Bharatanatyam mudras. In addition, deep CNN was used to classify the poses without extracting the features. In addition, complex environments and challenges with dynamic perception make dynamic hand gesture recognition a difficult topic in human-robot communication. To overcome this issue, dynamic hand gesture recognition and data-glove-based hand gesture recognition scheme are combined to make it 3D hand gesture estimation technique in Gao *et al* [11]. To increase the recognition accuracy of dynamic hand gestures, deep neural network and data fusion scheme were used.

In Caramalau *et al* [12], Bayesian approximation based deep learning architecture is presented for 3D HPE. It explores and analyzes two types of uncertainties that affect the data or learning ability. Also, compare

with the standard evaluator on other popular schemes. Graph Neural Network based HPE technique was proposed in Leng *et al* [13]. To estimate the depth image, adjacency matrix is designed to dynamically adjust the topography of the hand map during messaging and aggregation. Then, to obtain a consistent hand pose, a tremor compensation module is applied on the constraint adjacency matrix that utilizes the barrier between the control points and the tremor arm pose.

Pose detection has received much attention in the fields of human sensitivity and artificial intelligence. Most of the current studies use traditional machine learning classifiers to identify posture. Yet, these methods do not work well to accurately detect postures. Therefore, Liaqat *et al* [14] proposed a hybrid method based on different classifiers of machine learning such as logistic regression (KNN), Naive Bayes, support vector machine, decision tree, linear discrete analysis, random forest, and deep learning classifiers to identify posture detection. Therefore, to enhance the mudra pose estimation the different networks and bio-inspired optimization algorithms are utilized in the proposed method.

3. Overview of proposed methodology

3D HPE has recently made significant progress, with CNNs playing a key role. However, most existing CNN-based HPE techniques are reliant on the training package, and labeling 3D hand poses on training data is time consuming and costly. Therefore, a new method is proposed for 3D HPE based on EGPNet with 3D pose correction network. Initially, feature extraction is done by DetNet with improved EGPNet. The EGPNet is enhanced by the Smish activation function. After the feature extraction, the HPE is performed based on 3D pose correction network. Additionally, to enhance the estimation performance, Coot optimization algorithm is used to optimize the error between estimated and ground truth hand pose. Figure 1 demonstrates the proposed methodology.

3.1. Network backbone-DetNet

The goal of the method is to estimate the 3D hand pose from a single image in a 3D estimation structure. In this paper, a backbone network of DetNet [15] is used for down-sampling operations. Moreover, to maximize the receptive fields in the DetNet, dilated 1×1 conv layer is used rather than down-sampling processes after stage 3 as mentioned in the figure 2. To get a lot of semantic information, DetNet has sufficient deep layers and it continues a larger spatial resolution than other backbone networks, which can detect the image boundary. To get high-level semantic features, the starting step is reduced from 4 to 2 for larger receiving fields [16].

To extract multi-scale characteristics, DetNet utilizes five stages (2–6). Because the convolution is close to the input and the receptive field is too tiny, the level 1 is thrown, and the level 2 is meant to extract the edge. This convolutional features are indicated into $f^D \in R^{W \times H \times C}$, where R represents the convolutional features and H, W, C indicates height, width and channel number of each R correspondingly, $f^D \in f_1^D, f_2^D, f_3^D, f_4^D, f_5^D, f_6^D$

$$Y_i^D = \sigma (F_i^{\text{out}} (f_i^D; \theta_i^D)), i \in \{3, 4, 5, 6\}. \quad (1)$$

Here, Y_i^D represents DetNet output features, f_i^D indicates the i th path DetNet features, $F_i^{\text{out}} \in R^{W \times H \times 1}$ refers to the output features, θ_i^D indicates the convolution parameter of F_i^{out} , $\sigma(\cdot)$ refers to Sigmoid function. The DetNet conv features expression of equation (1) is used to learn the backbone features.

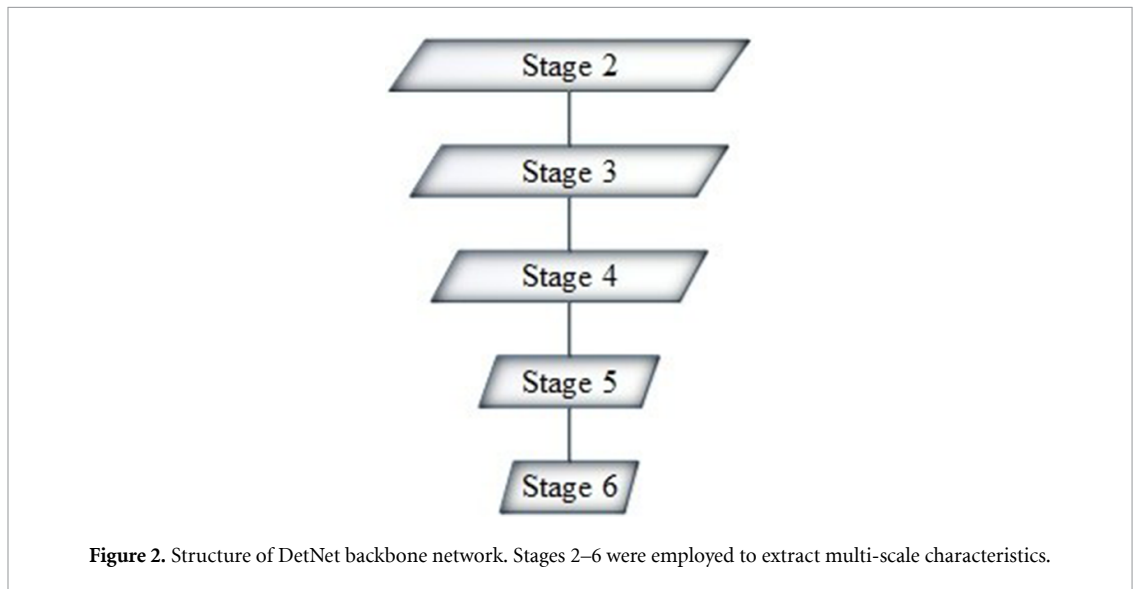
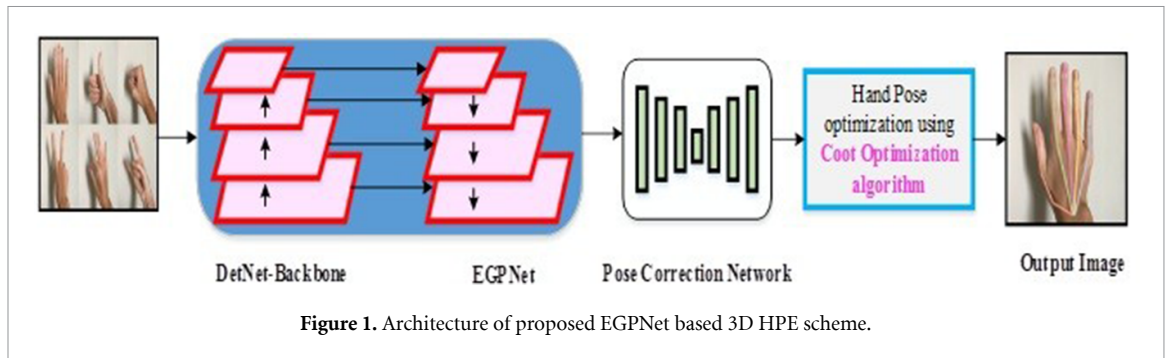
3.2. EGPNet

The DetNet is used as backbone for the estimation network. This research aims to utilize the EGPNet as a feature extractor for DetNet.

The residual blocks of various convolution features are 2, 3, 4, and 5 as $C_1, C_2, C_3,$ and C_4 , respectively. To create heatmaps for keypoints, 3×3 convolution filters are applied to C_2 to C_5 layers. The layers C_2 and C_3 have high spatial resolution shallow features for localization but less semantic information. In contrast, deep feature layers of C_4 and C_5 have high semantic information but less spatial resolution because of pooling operation and strided convolution function. Hence, to sustain semantic information and spatial resolution, feature pyramid network (FPN) structure is integrated for the feature layers. FPN enhances the deeply supervised information.

Figure 3 introduced EGPNet is similar to the FPN for hand pose keypoint estimation. Slightly different from FPN, 1×1 conv kernel is applied before each element-wise addition in the up-sampling process and the layer followed by a Smish function as an activation layer. The Smish inherits the nonmonotonic properties of the Logish function. The Smish function is mostly suitable for machine learning networks which can be expressed in equation (2) [17],

$$f(x) = x \cdot \tanh [\ln (1 + \text{sigmoid}(x))]. \quad (2)$$



sigmoid(x) is used to reduce the range of values, and the logarithmic function is used to obtain a smooth curve and a flat trend. Smish multiplies its tanh function by x simultaneously, thus expressing the regulatory ability for negative inputs; conversely, positive values become simple linear expressions. Therefore, the architecture called EGPNet. Based on the DetNet, EGPNet can effectively detect vital keypoints such as the eyes but may not be able to accurately detect the position of the hips. Localization of key points such as the hip usually requires more environment than the proximal aspect feature.

3.3. Estimation network

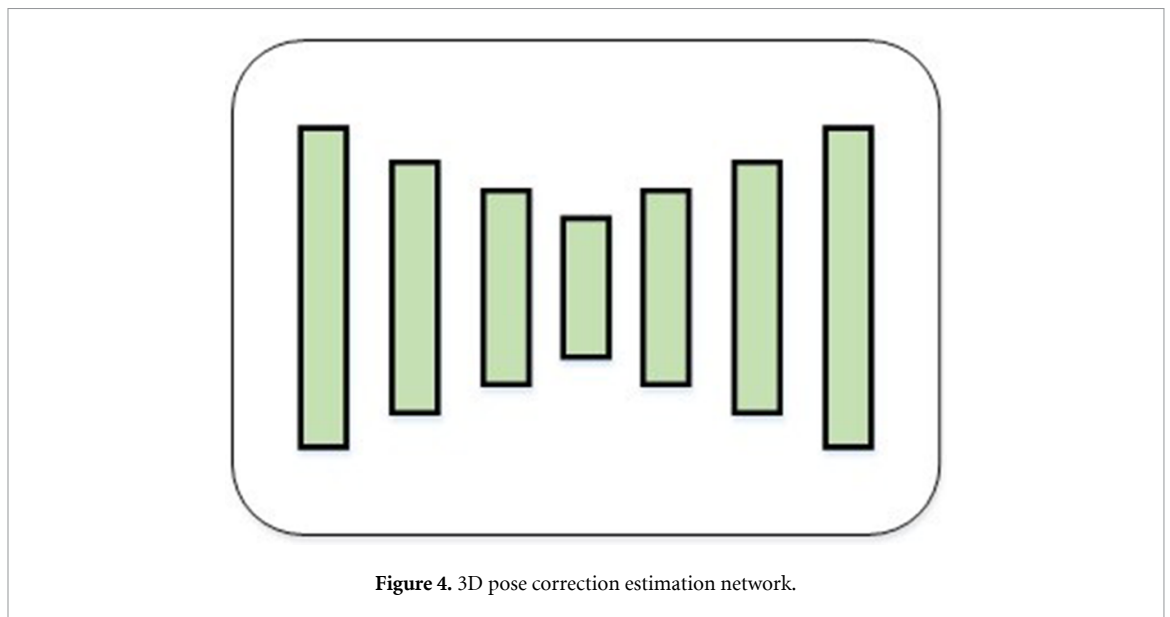
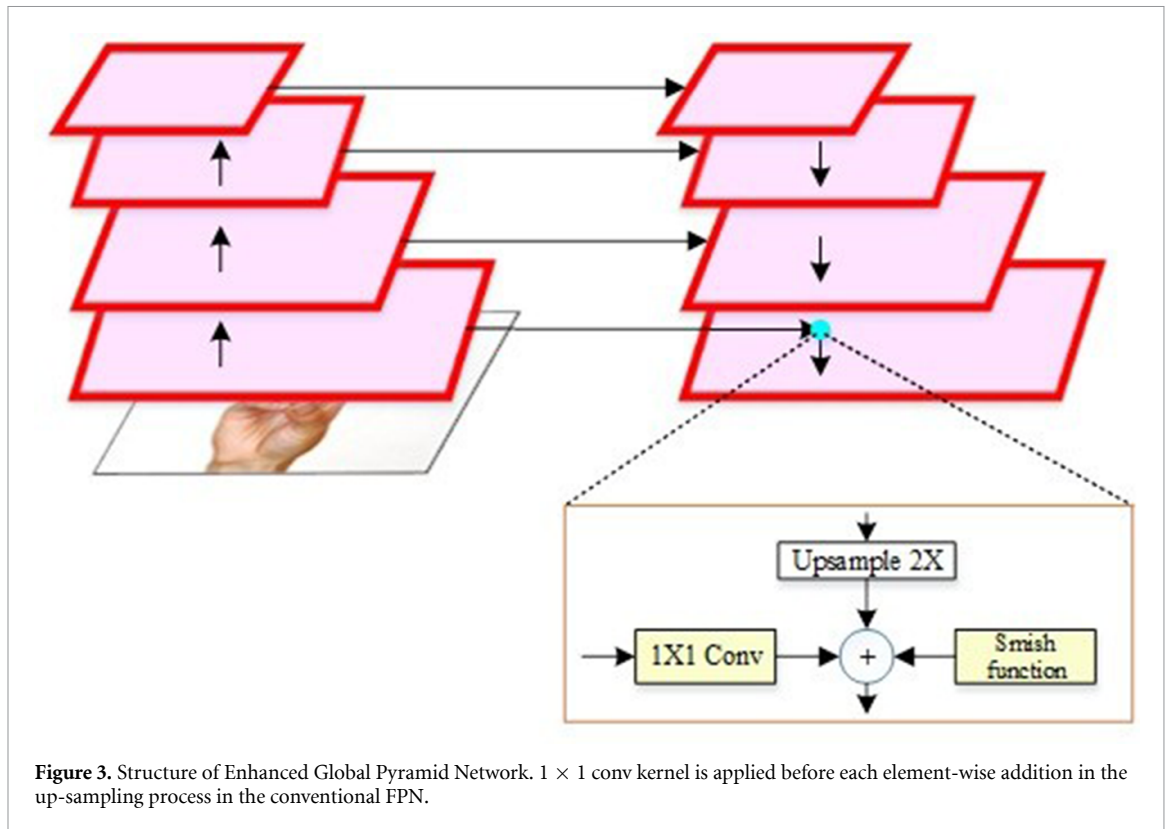
The Pose Correction Network depicted in figure 4 will be used to tune the initial keypoint heatmap estimation $k_i(c)$, which provides adjustable keypoint heatmaps. At first, the extracted feature $\Psi_i(p, c, n)$ is used as the input of pose correction estimation network. It consists of five parallel 3×3 conv layers and different dilation rates as $d \in 3, 6, 9, 12, 15$. This calculation provides five sets of offsets for the next kernel layer of the deformable conv layer. The offsets are calculated based on equation (3) [18],

$$\Phi_{i(p)} \mathbf{M}_{\Psi_{i(p)}} \frac{\text{stack of } 3 \times 3}{\text{residual blocks convlayer}} \frac{d}{d} O_{i,d} \tag{3}$$

Here, keypoint heatmaps and feature tensor is denoted into $\Phi_i(p)$ and $\Psi_i(p)$ it will enhance the pose estimation.

The dilation rate varies depending on the size of the receptive field while large dilation rate upturns the receptive field. The local appearance is focused by a less dilation rate which is sensitive to capture subtle motion contexts. Small expansion rate emphasizes the local appearance, which is sensitive to capture subtle operating environments. On the contrary, the use of a high dilation rate allows global representations to be encrypted and capture relevant information to the larger spatial scope. Additionally in offset computation, the keypoint heatmaps is integrated into similar conv layers and getting five sets of masks m_d

$$\Phi_{i(p)} \mathbf{M}_{\Psi_{i(p)}} \frac{\text{stack of } 3 \times 3}{\text{residual blocks convlayer}} \frac{d}{d} m_{i,d} \tag{4}$$



The offset O and mask m are independent parameters of dilation convolution structure. A mask M_d refers to the weight matrix for a convolution kernel. This module is implemented through the various dilation rates based on merged keypoint heatmaps $\Phi_i(p)$, kernel offsets $O_{(i,d)}$ and the masks $m_{(i,d)}$ and outputs a pose heatmap for image i at dilation rate d .

Then the final pose prediction for image i is achieved from the five outputs for five dilation rates are summarized which is in equation (5),

$$\sum_{d \in \{3,6,9,12,15\}} K_{i,d} \xrightarrow{\text{normalization}} K_i(c). \tag{5}$$

Finally, the predicted pose is achieved. To enhance the 3D HPE network, the bio-inspired optimization algorithm is used. A set of normalized points is fed into the pose regression problem as $X = x_{i(i=1)}^N = p_i, n_{i(i=1)}^N$ and outputs estimates the pose \ddot{P} where p_i is the 3D coordinate of the point and n_i is normal 3D surface. Equation (6) describes a regression function r_f ,

$$\hat{P} = r_f(X, \theta_r) \quad (6)$$

where, θ_r is the trainable parameters of r_f . In the proposed method, Coot optimization algorithm is applied to optimize the parameters θ_r in order to reduce the error among the estimated hand pose \hat{P} and ground truth hand pose P

$$\text{obj}_{\text{fun}} = \hat{P} - P. \quad (7)$$

3.4. Coot optimization algorithm

Coots are little aquatic birds that belong to the Rallidae family of rails. The Coots water surface behavior is used to build the optimization algorithm [19]. Coots travel at angles in the direction of their movement and for surf scoters, will be the chase zone. In optimization, considers the four moves such as

- Random movement
- Chain movement
- Based on the group leaders adjusting the location
- leader movement

3.4.1. Random movement

The population is randomly initialized into $\vec{t} = \vec{t}_1, \vec{t}_2, \dots, \vec{t}_n$. The random population is frequently computed by the objective function. The population is generated by equation (8),

$$C_L = \text{rand}(1, g) \times (u_b - l_b) + l_b \quad (8)$$

where, Coot location is denoted into $C_L(i)$, problem dimension, lower and upper boundary is indicated as d , l_b and u_b , respectively. In the search space, Coot move towards the random location. This Coot movement discovers various areas of the search space. This can be avoid the local optimal stuck issues and new location is updated by equation (9),

$$C_L(\text{new}i) = C_L(i) + A \times r_1 \times (C_L(i) - C_L(\text{initial})) \quad (9)$$

$$A = 1 - T \frac{1}{\max_i} \quad (10)$$

where, current iteration and maximum iteration is denoted into T and \max_i , respectively. r_1 represents the random number in the range of 0–1.

3.4.2. Chain movement

In the group of Coot, one leader is leading the other Coots in front of the group and other Coots adjusting its location and move towards them. Consider the average position of the leaders, Coots can update its location their position. To choose the leader use following equation (12),

$$I = 1 + (i \text{MOD} L_n) \quad (11)$$

where, leader's index number is I , number of leaders is L_n , and current Coot index number is denoted as i . Based on leader location update the position by equation (13),

$$C_L(i) = L_I(I) + 2 \times r_2 \times \cos(2\pi R) \times L_I(I) - C_L(i) \quad (12)$$

where, $L_I(I)$ is chosen leader location, r_2 is a random number and R denotes random number lies in the range of $-1-1$.

3.4.3. Leader movement

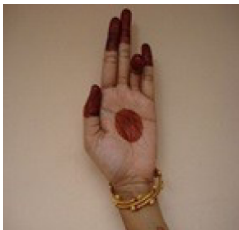


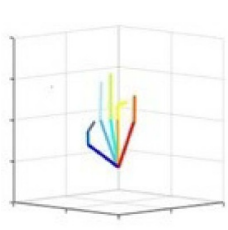







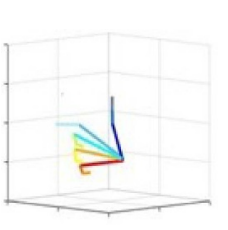



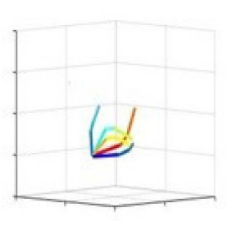
The group should be engaged towards the best optimal location, so the leaders should update its position towards the optimal position

$$L_I = \begin{cases} B \times r_3 \times \cos(2\pi R) \times (\text{best} - L_I(I)) & r_4 < 0.5 \\ B \times r_3 \times \cos(2\pi R) \times (\text{best} - L_I(I)) & r_4 \geq 0.5 \end{cases} \quad (13)$$

$$B = 2 - T \frac{1}{\max_i}. \quad (14)$$

$\cos(2\pi R)$ searches with various radius around the best search agent in order to discover a better position around the search agent. Finally optimal solution is obtained by the Coot optimization approach.

Table 1. 3D hand pose estimation for Bharatanatyam dataset.

Bharatanatyam			
Original image	2D pose estimated	Cropped image	3D estimated
			
			
			
			

4. Result and discussion

The proposed 3D hand mudra estimation is experimentally verified on four types of datasets such as Bharatanatyam, yoga, Kathakali and sign language image sets. The experiments have been carried out to establish the effectiveness of the proposed EGPNet with estimation network for mudra detection technique in Google Colab using python programming with core i3 processor and 4GB RAM system. The proposed method is analyzed based on the collected datasets.

4.1. Bharatanatyam dataset

The dataset is prepared from various websites and Bharatanatyam dance videos. Bharatanatyam dance mudras are available on the websites of Indian art and culture. During training phase Anjali, Dhyana, Gyan, Kubera, Prana, Rudra, Surya, and Vayu dance mudras are used to estimate the mudra pose.




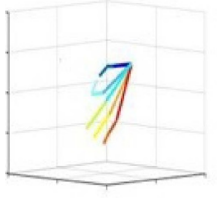


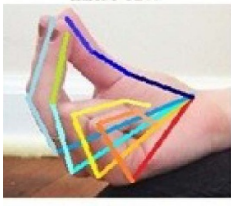
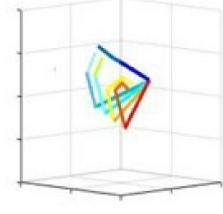


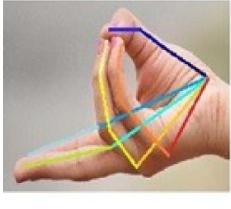
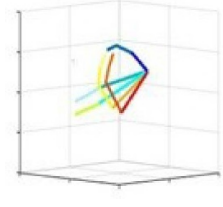



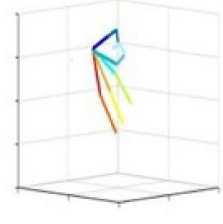
4.2. Yoga dataset

The yoga images are collected from various websites. The collected images include both phases as training and testing phase. For mudra estimation, yoga mudra images of Alapadma, Anjali, Chakra, Chandrakala, Garuda, Pasha, Pataka, Pushpaputa, Shivalinga, Simhamukha, and Thripathaka are used.

4.3. Kathakali dataset

Kathakali hand mudra dataset is available in [20]. To training and testing the dataset, 2272 and 200 images are used. The dataset of Kathakali Hand Mudras are containing five mudras such as Pathaaka, Mudraakhyam, Katakam, Mushti, and Kartharee Mukham. Hand image are annotated in Pascal visual object classes (VOC) format.

Table 2. 3D hand pose estimation for yoga dataset.

Yoga			
Original image	2D pose estimated	Cropped image	3D estimated
			
			
			
			

4.4. Sign language dataset [21]

The data set consists of images of alphabets from American Sign Language that indicate different classes. The dataset contains 87 000 images with a resolution of 200×200 pixels make up the training data set. There are 29 classes in total, with 26 of them dedicated to the letters A to Z. To encourage the use of real-world test images, the test data set contains only 29 images.

These images are preprocessed to smooth the pixel values. During data preprocessing, the images are cropped to remove the inappropriate background and to ensure that the hands are situated in the center of the images. Then the cropped images are then scaled into 256×256 resolution. Feature extraction section is instigated to extract features from the mudra images. Feature of each mudra is labeled to recognize the mudras and keypoints are estimated by the 3D pose correction network. The estimated Bharatanatyam mudras are illustrated in table 1. At first original mudra images are provided and after 2D and 3D estimated pose are given. Additionally, the mudras are recognized and displayed the names in top of the image.



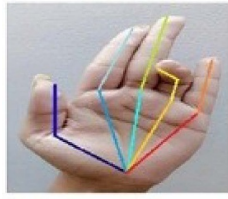
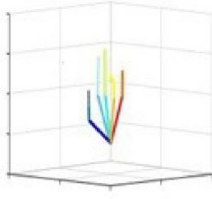

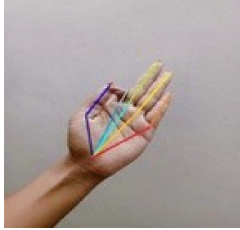
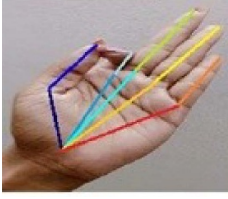
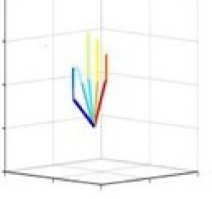


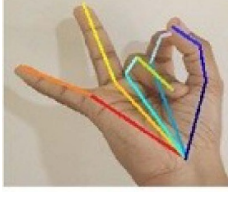
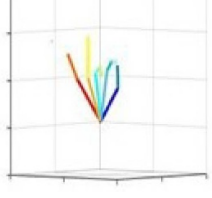


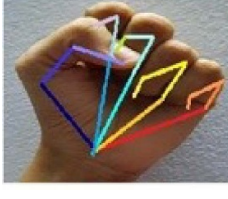
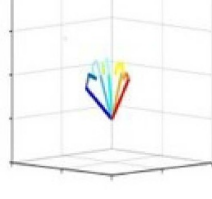
Table 2 shows the 3D HPE result of yoga dataset. From the table we can observe the original image, 2D pose and 3D estimated output image. Additionally, the yoga poses are recognized and displayed the names in top of the image for example, the poses Gyan, Kubers, Prana and Vayu.

Table 3 shows the 3D HPE result of Kathakali dataset. The original Kathakali image, 2D pose and 3D estimated output image results are observed from the table 3. Also, the Kathakali mudras are recognized and shown the mudras names in top of the image for example, Pathaka, Mudraakhyam, Katakam, and Mushti.

Table 4 shows the 3D HPE result of sign language dataset. The original sign language image, 2D pose and 3D estimated output image results are observed from the table 4. Also, the sign language are recognized and shown the sign names in top of the image for example, B, V, F and W.

The proposed model is trained with stochastic gradient descent (SGD) on a graphical processing unit (GPU) which means using two images per GPU. The proposed network is trained for 100 epochs with 0.000 05 initial learning rates. The momentum is set as 0.9. The Adagrad optimizer is used as optimizer. The

Table 3. 3D hand pose estimation for Kathakali dataset.

Original image	Kathakali		
	2D pose estimated	Cropped image	3D estimated
		 Pathaka	
		 Mudraakhyam	
		 Katakam	
		 Mushti	



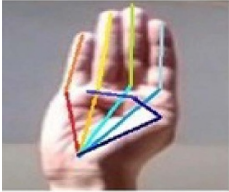
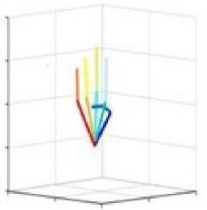



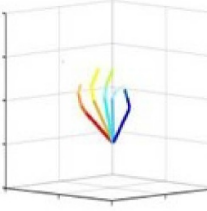

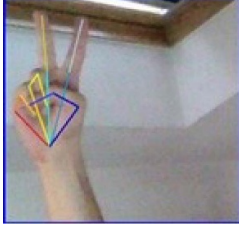

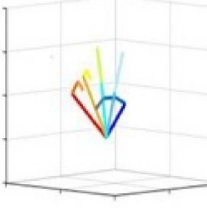

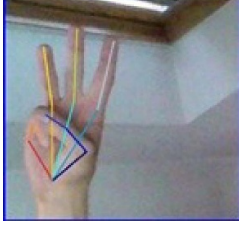
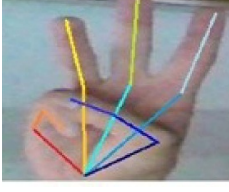
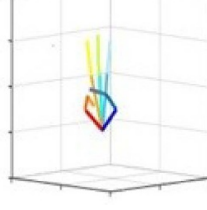
prior probability is set into 0.01. The pyramid and classification feature size is set as 256. The training sample region is denoted as R and radius is set into 1.5 and dropout rate is 0.5. In the sampling region each location point is annotated by the type of material relevant to the training. Additionally, the horizontal flipping is used for datasets to increase training samples.

The training loss evaluation is demonstrated in figure 5. The loss curve is plotted between epoch and loss value. Initial loss value of proposed method is higher and falls efficiently with the upturn of epochs. If number of epoch is increases loss value is also decreases. The loss value is very low when the sample is in 240 epochs of training. The validation loss is less than 0.25 compared to training loss. Hence, the proposed network model achieves high accuracy rate and a less loss value. To reduce over-fitting on training data, dropout regularization method is used for preventing intricate co-adaptations on training data. It is an extremely effective method for using neural networks to average models. The proposed network training process accuracy is illustrated in figure 6. From the observation the training process is stable for the proposed network and quickly converges. Based on the dataset, all networks are operating in the accuracy curve. Training accuracy comparison is performed between training and validation dataset. From the accuracy analysis, the proposed network quickly given better results and training process is more stable.

4.5. Performance evaluation

To evaluate the 3D HPE performance, the measures of mean EPE, EPE median and AUC are used. The average Euclidean distance between anticipated and ground-truth joints is measured by EPE mean metric. AUC on the percentage of correct keypoints curve based on certain error thresholds. As the boundary for the provided curve, the area of the curve can be determined with regard to the various axes. Either the x -axis or the y -axis can be used to determine the AUC. The EPE mean is the average obtained by dividing the total number of numbers by the sum of all the numbers, whereas the EPE median is the midway value obtained by numerically arranging the given numbers from smallest to largest [22]. The EPE median is significant

Table 4. 3D hand pose estimation for sign language dataset.

Sign language			
Original image	2D pose estimated	Cropped image	3D estimated
			
			
			
			

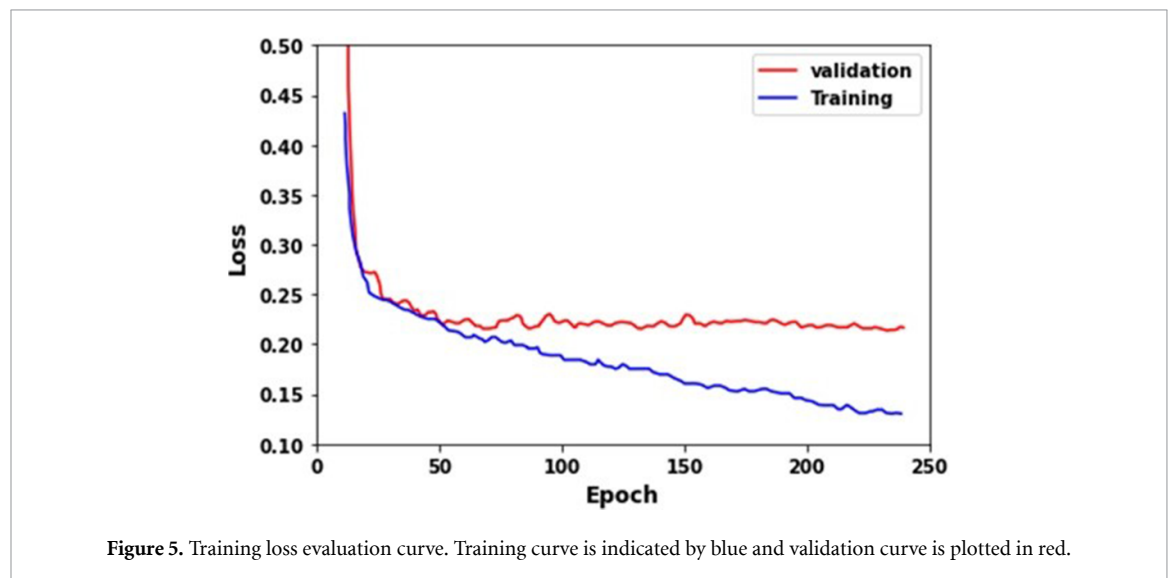
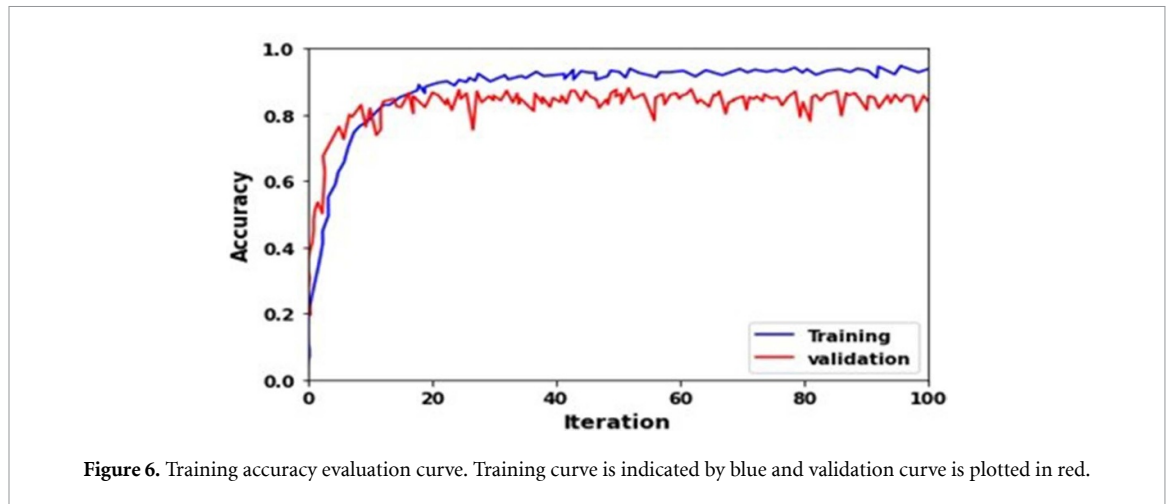


Figure 5. Training loss evaluation curve. Training curve is indicated by blue and validation curve is plotted in red.

because it helps us determine where the middle value in a dataset is located. When a distribution is skewed and/or has outliers, the median is typically easier to determine than the mean. The considered performance matrices are evaluated on the testing dataset.

Table 5 shows the comparative analysis of different optimization algorithms. To analyze the performance, the proposed method is compared with different optimization algorithms with Smish activation function. The proposed Coot-Smish achieves better results than the Smish with genetic algorithm (GA) (0.78),

**Table 5.** Comparative analysis of different optimization algorithms.

Optimization + activation	AUC	EPE median	EPE mean
GA-Smish	0.78	12.84	22.34
IPSO-Smish	0.82	10.68	20.20
PSO-Smish	0.88	8.25	19.73
Coot-Smish	0.95	7.48	17.06

Table 6. Comparative analysis of different activation functions.

Activation function	AUC	EPE median	EPE mean
Sigmoid	0.76	11.33	21.15
Relu	0.80	9.34	19.69
Swish	0.82	8.14	18.90
Smish	0.92	7.05	17.60

Table 7. Comparative analysis of with and without optimization.

	AUC	EPE median	EPE mean
With optimization	0.91	7.56	17.8
Without optimization	0.74	9.86	21.44

improved particle swarm optimization (IPSO) (0.82) and particle swarm optimization (PSO) (0.88) algorithms in AUC comparison. Similarly, for EPE median and EPE mean analysis, Coot and Smish combination achieves higher performance than the other methods.

These metrics validates the overall performance of each estimated joint and hand pose which indicates the performance of a given estimation technique. Table 6 shows the comparative analysis of different activation functions. While avoiding issues brought on by neural networks itself, optimization approaches will speed up calculations. The neural network model's ability to learn the intricate nonlinear relationship depends critically on the activation function. The neural network must reflect the nonlinear properties in order to work. Therefore, the introduced Smish function is compared with Sigmoid, Relu and Swish activation functions. The pre-trained network, which was trained and tested on Smish, performs well with a mean EPE of 17.60 mm and median EPE 7.05 mm.

To enhance the estimation accuracy, the Coot optimization algorithm is used to tune the trainable parameters. Therefore, to show the effectiveness of the optimization approach, the proposed method is compared to with and without optimization approach which is shown in table 7 and figure 7. When the optimization approach is not applied, the method achieves lower performances in AUC, Median EPE and Mean EPE as 0.74, 9.86 and 21.44, respectively. However, with optimization the method achieves higher performances in AUC, Median EPE and Mean EPE as 0.91, 7.56 and 17.8, respectively.

The 3D HPE performances are computed in pixels. Since there is no works on 3D HPE from yoga and Bharatanatyam images yet, we cannot compare to different approaches. To still relate our results coarsely to existing works, we compare the proposed network is compared with ResNet-50, ResNet-101, EfficientNet-b2

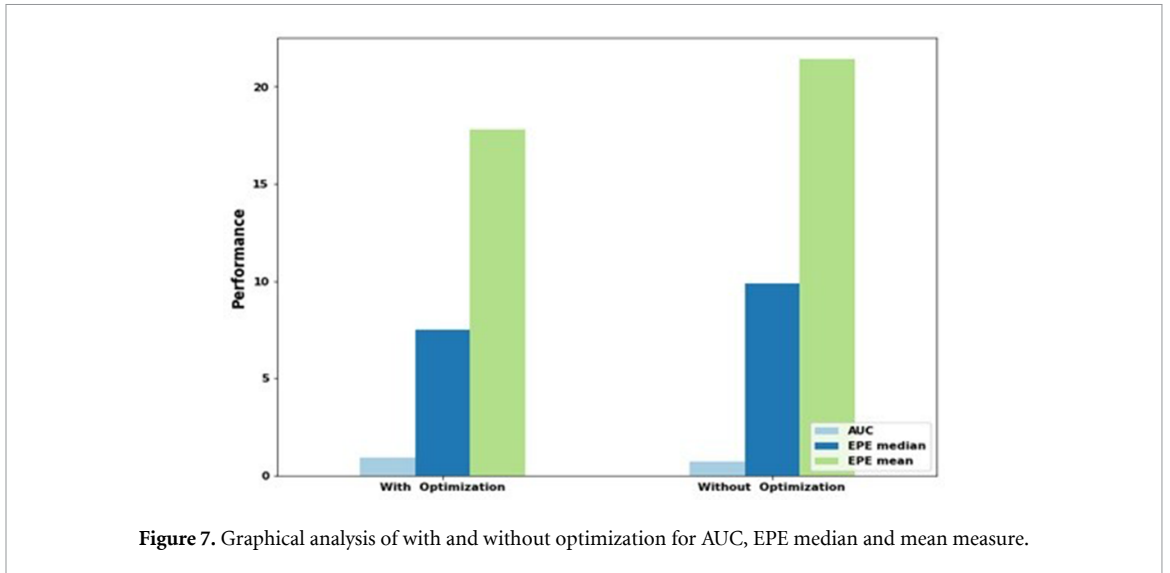


Figure 7. Graphical analysis of with and without optimization for AUC, EPE median and mean measure.

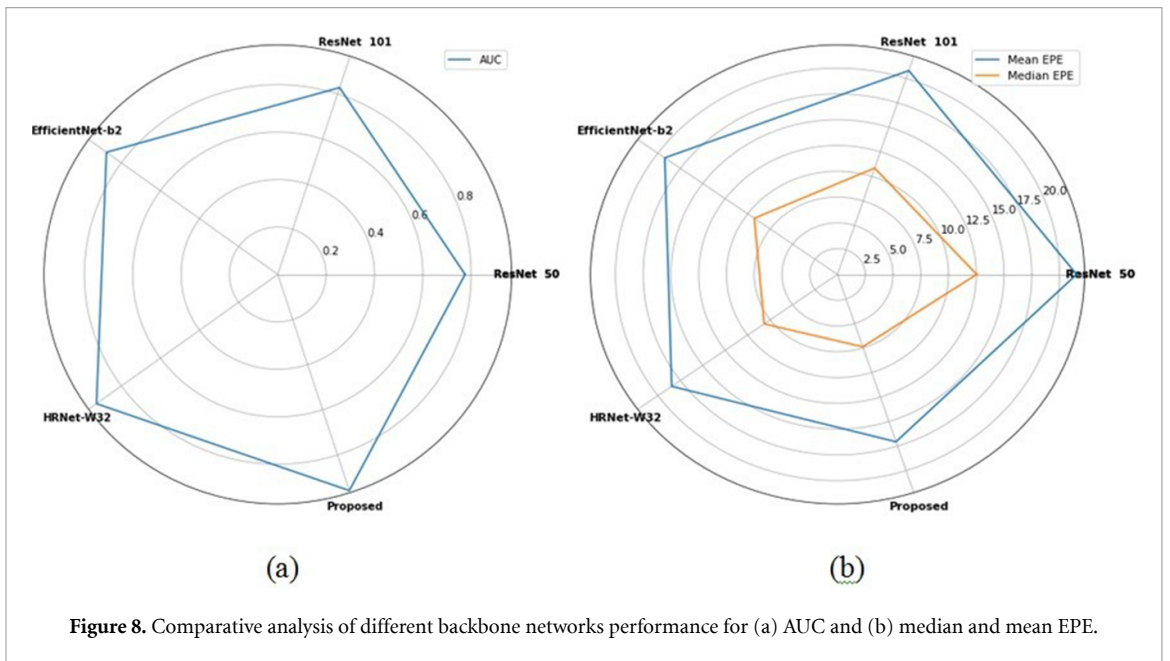


Figure 8. Comparative analysis of different backbone networks performance for (a) AUC and (b) median and mean EPE.

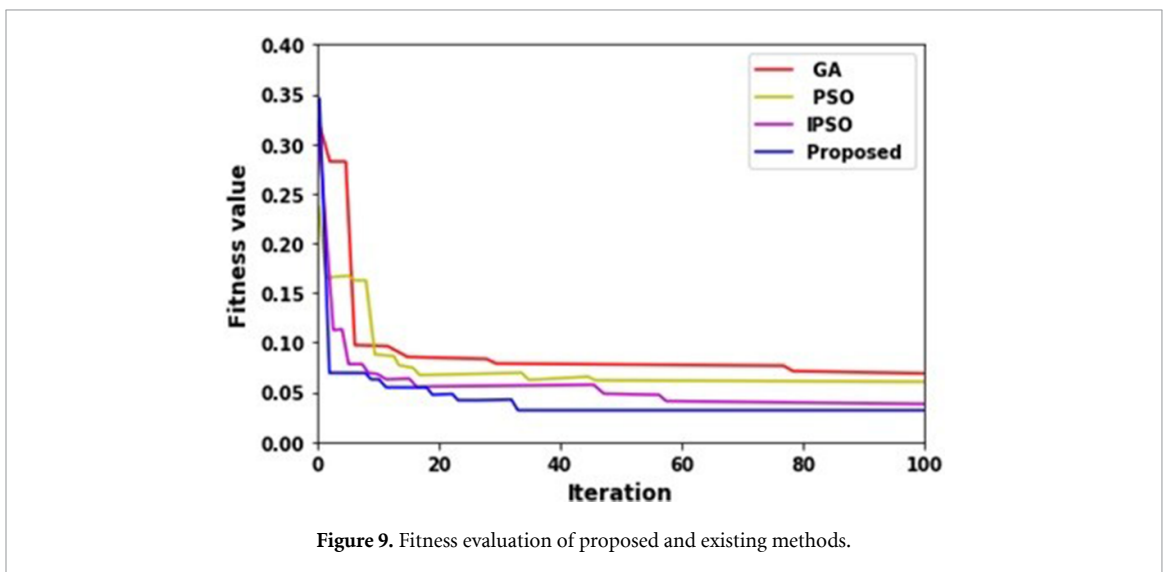


Figure 9. Fitness evaluation of proposed and existing methods.

and HRNet-W32 for 3D HPE. From figure 8, DetNet backbone achieves better performance in terms of AUC, mean EPE and median EPE as 0.95, 7.38 and 17.06, respectively.

The fitness function of proposed method is to minimize the error among the estimated and ground truth hand pose. Therefore, the fitness analysis is provided in figure 9. To evaluate the fitness function, the proposed Coot optimization compared with genetic algorithm [23], IPSO [24] and PSO [25] algorithms. From the graphical analysis, Coot optimization achieves very less error value which means high fitness range as 0.025.

5. Conclusion

We offer a unique optimization-based 3D HPE methodology is to accurately address the problem of hand posture estimation in this work. The proposed 3D HPE scheme is evaluated on yoga, Kathakali, Bharatanatyam and sign language RGB image datasets. The images are having different level features so the hybrid network as DetNet and Enhanced Global Feature Pyramid network is designed to extract multi-scale multi-level feature maps. The Smish activation function is provided to enhance the FPN network. For 3D HPE, the 3D pose correction network is introduced. To enhance the estimation accuracy, the Coot optimization algorithm is used to optimize the error between estimated and ground truth hand pose image. The proposed scheme validated efficient results on four datasets. The proposed method is evaluated based on the performances measures of AUC, median EPE and mean EPE. Also, the proposed method is compared with various existing methods such as, EfficientDet, ResNet-50, ResNet-100 and etc. The proposed method increases the 3D hand pose mudra estimation accuracy.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: www.kaggle.com/datasets/sauravsm/kathakali-mudra-dataset.

ORCID iDs

Pallavi Malavath  <https://orcid.org/0000-0003-4180-6062>

Nagaraju Devarakonda  <https://orcid.org/0000-0003-4864-8482>

References

- [1] Chatzis T, Stergioulas A, Konstantinidis D, Dimitropoulos K and Daras P 2020 A comprehensive study on deep learning-based 3D hand pose estimation methods *Appl. Sci.* **10** 6850
- [2] Baek S, Kim K I and Kim T K 2019 Pushing the envelope for rgb-based dense 3D hand pose estimation via neural rendering *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 1067–76
- [3] Chen X, Wang G, Guo H and Zhang C 2020 Pose guided structured region ensemble network for cascaded hand pose estimation *Neurocomputing* **395** 138–49
- [4] Mallick T, Das P P and Majumdar A K 2019 Posture and sequence recognition for Bharatanatyam dance performances using machine learning approach (arXiv:1909.11023)
- [5] Naik A D and Supriya M 2021 Classification of Indian classical dance 3D point cloud data using geometric deep learning *Computational Vision and Bio-Inspired Computing* (Singapore: Springer) pp 81–93
- [6] Yang L, Chen S and Yao A 2021 SemiHand: semi-supervised hand pose estimation with consistency *Proc. IEEE/CVF Int. Conf. on Computer Vision* pp 11364–73
- [7] Huang L, Tan J, Liu J and Yuan J 2020 Hand-transformer: non-autoregressive structured modeling for 3D hand pose estimation *European Conf. on Computer Vision* (Cham: Springer) pp 17–33
- [8] Oberweger M, Wohlhart P and Lepetit V 2015 Hands deep in deep learning for hand pose estimation (arXiv:1502.06807)
- [9] Ge L, Cai Y, Weng J and Yuan J 2018 Hand pointnet: 3D hand pose estimation using point sets *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 8417–26
- [10] Priya B G and Arulselvi M 2019 Deep learning for human pose classification using multi view dataset *Int. J. Recent Technol. Eng.* **8** 325–8
- [11] Gao Q, Chen Y, Ju Z and Liang Y 2021 Dynamic hand gesture recognition based on 3D hand pose estimation for human-robot interaction *IEEE Sens. J.* **22** 17421–30
- [12] Caramalau R, Bhattarai B and Kim T K 2021 Active learning for Bayesian 3D hand pose estimation *Proc. IEEE/CVF Winter Conf. on Applications of Computer Vision* pp 3419–28
- [13] Leng Z, Chen J, Shum H P, Li F W and Liang X 2021 Stable hand pose estimation under tremor via graph neural network 2021 *IEEE Virtual Reality and 3D User Interfaces (VR)* (IEEE) pp 226–34
- [14] Liaqat S, Dashtipour K, Arshad K, Assaleh K and Ramzan N 2021 A hybrid posture detection framework: integrating machine learning and deep neural networks *IEEE Sens. J.* **21** 9515–22
- [15] Li Z, Peng C, Yu G, Zhang X, Deng Y and Sun J 2018 Detnet: a backbone network for object detection (arXiv:1804.06215)
- [16] Tan Z and Gu X 2021 Depth scale balance saliency detection with connective feature pyramid and edge guidance *Appl. Intell.* **51** 5775–92
- [17] Wang X, Ren H and Wang A 2022 Smish: a novel activation function for deep learning methods *Electronics* **11** 540

- [18] Liu Z, Chen H, Feng R, Wu S, Ji S, Yang B and Wang X 2021 Deep dual consecutive network for human pose estimation *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* pp 525–34
- [19] Naruei I and Keynia F 2021 A new optimization method based on COOT bird natural life model *Expert Syst. Appl.* **183** 115352
- [20] Saurav S *Kathakali Mudra Dataset* (available at: www.kaggle.com/datasets/sauravsm/kathakali-mudra-dataset)
- [21] Akash *ASL Alphabet* (available at: www.kaggle.com/datasets/grassknotted/asl-alphabet)
- [22] Zimmermann C and Brox T 2017 Learning to estimate 3D hand pose from single rgb images *Proc. IEEE Int. Conf. on Computer Vision* pp 4903–11
- [23] Mirjalili S 2019 Genetic algorithm *Evolutionary Algorithms and Neural Networks* (Cham: Springer) pp 43–55
- [24] Ji Y, Liew A W C and Yang L 2021 A novel improved particle swarm optimization with long-short term memory hybrid model for stock indices forecast *IEEE Access* **9** 23660–71
- [25] Bansal J C 2019 Particle swarm optimization *Evolutionary and Swarm Intelligence Algorithms* (Cham: Springer) pp 11–23