Iberian Journal of Applied Sciences and Innovation

2022 vol 2, Issue3

Hand gesture detection by genetic algorithm and multilayer perceptron

Fatemeh Mirzaei Asl
Department of Computer Engineering
Khorramabad Branch, Islamic Azad University
Khorramabad, Iran
f_mirzaei9920@yahoo.com

Saman Shojae Chaeikar
Sydney International School of Technology and Commerce
Sydney, Australia
samans@sistc.edu.au
Saeid Yazdanpanah
Department of Computer Engineering
Khorramabad Branch, Islamic Azad University
Khorramabad, Iran
yazdanpanah.saeid@iau.ac.ir

Arman Roohi
Sydney International School of Technology and Commerce
Sydney, Australia
armanr@sistc.edu.au

Abstract— **Recently, controlling robots through remote user hand motions has turned into an important research area in robotics. Robots normally employ a combination of image processing and learning techniques to detect hand poses, and metaheuristic techniques that follow the natural rules or behavioral models of living things can help to enhance their performance. This paper proposes a supervised learning method that applies genetic algorithm principles on a multilayer perceptron to improve the performance of the current MLP-based algorithms for hand pose detection. The method is trained and evaluated on a dataset consisting of data acquired from 300 users. RMSE benchmark reports an index value of 0.0424, and the experiments show a sensitivity of 93.42% and accuracy of 92.27% – 10.25% and 6.65% improvement compared to the MLP implementation.**

Keywords— hand pose detection, finger bent angle, genetic algorithm, MLP

I. INTRODUCTION

Learning and recognition are two of the most important capabilities of humankind. A major part of recognition and perception relies on sight. Seeing the surrounding environment, the human can stay alert and learn based on its visual experiences [1,2]. Correct understanding and decision-making in different conditions are the critical factors to survive. Although it seems that human object recognition is a trivial and basic task, the researchers are still struggling with implementing this feature in the manmade devices [3,4].

The image processing systems fall into two groups: image optimization and machine vision [5,6]. Image optimization encompasses the methods that enhance the quality of image, while machine vision algorithms try to understand the image content, especially for robotic applications [7,8,9]. An important application of image processing is hand motion detection. It analyzes low-level features such as color and edge to deliver human-level capabilities in analysis, classification, and content recognition [10,11].

Recently, outstanding advances have been made in hand motion detection. The image processing techniques perform segmentation and feature extraction on the input images and detect the hand pose, fingertips, and hand palm in real-time. These features are then used to compute the bending angle of fingers for robots [12,13,14]. During the past twenty years, the researchers used color or wired gloves which are equipped with implementable sensors to detect hand pose. The skin-color-based analysis does not require a paper cover or sensors in order to detect fingers, and works fast, accurately, and in real-time [15,16,17].

Vision-based systems can be utilized to control a remote robotic hand by sending the human hand pose signals in real-time and then imitate the pose in the remote area for a set of critical actions [18,19]. To lift an object, the user bends its fingers – virtually – to lead the robot to lift the real object. These systems first transform the frames into two-dimensional images and then apply segmentation and skin filter functions. In this area, the researchers have presented various methods that utilize machine learning techniques such as support vector machines [20,21], artificial neural networks [22,23], fuzzy systems [24,25], deep learning [26], and metaheuristic methods [27,28,29].

Modern practice for controlling robot movements is training it on a set of natural hand movements and then reducing the detection errors by metaheuristic techniques such as swarm algorithms, evolutionary algorithms, natural-inspired algorithms, or physical algorithms [30,31].

Metaheuristic is a class of algorithms in which the behavioral model of animals or natural phenomenon are used for optimum solutions to engineering problems. Examples of these algorithms include gray wolf optimization algorithm [32], seagull optimization algorithm [33,34], emperor penguin optimizer [35,36,37], spotted hyena optimizer [38], whale optimization algorithm [39], sailfish optimizer [40], and poor-and-rich optimization algorithm [41,42].

II. Background and Related Works

Nowadays, robots have undertaken a wide range of roles in industry, ranging from simple tasks such as parts transportation to more complicated ones like spot welding and montage. Supported by the artificial intelligence advances, the robots now have more human-like and intelligent reactions [43,44]. For interactive communication between the robots and the surrounding environment, they require to be aware of the objects and routes. A plot is enough to resolve this problem in a static environment, while in a dynamic atmosphere the robots must utilize various machine vision and decision-making techniques to interact [45]. A sample robotic arm for moving objects is shown in Figure I, and Figure II display a motion detector that imitates the user's actions.

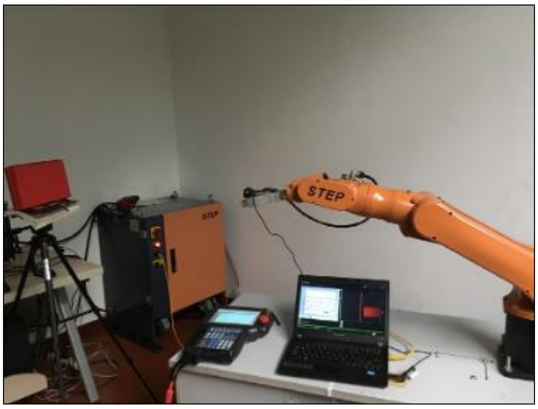

Figure I. A robotic arm for lifting objects [46]

Figure II. Hand motion detection by robots [47]

A common solution for user hand gesture recognition is using data-glove, hand-belt, and camera. Luzhnica *et al.* [48] used a hand-belt equipped with a gyroscope, accelerometer, and Bluetooth for hand gesture recognition. Hung *et al.* [49]

acquired the input required data from hand gloves. Another research has used Euclidean distance for analyzing twenty-five different hand gestures and employed a support vector machine for classification and controlling tasks [50]. In another effort [51], the researchers converted the RGB captured images to grayscale, applied a Gaussian filter for noise reduction, and fed the results to a classifier to detect the hand gesture.

Chaudhary *et al.* [52] used a normal Windows-based webcam for recording the user hand gestures. Their proposed method extracts the ROI from frames and applies an HSV-based skin filter on RGB images in particular illumination conditions. The model analyzes hand direction according to pixel density in the image, to help the fingertip detection process. It also detects the palm hand based on the number of pixels located in a 30×30-pixel mask on the cropped ROI image. The model has been implemented by a supervised neural network based on Levenberg–Marquardt algorithm and uses eight thousand samples for all fingers. The architecture of the model has five input layers for five input finger poses, and five output layers for the bent angle of the fingers. Marium *et al.* [53] extracted the hand gesture, palm, fingers, and fingertips from webcam videos using the functions and connectors in OpenCV.

Simion *et al.* [54] researched finger pose detection for mouse control. Their model, in the first step, detects the fingertips (except thumb) and palm hand. Its second step is to calculate the distances between pointing and middle fingers to the center of the palm, and the distance between the tips of pointing and middle fingers. The third step computes the angles between the two fingers, between the pointing finger and x-axis, and between the middle finger and x-axis. The researchers used six hand poses for mouse movements including right/left click, double click, and right/left movement.

This paper proposes a novel method for controlling robot hand movements based on the bending angle of human fingers by applying genetic algorithm principles on a multilayer perceptron. It is assumed that robot hand joints have the same degree of freedom as human joints. The process steps include detecting the palm center and fingertips in real time, defining the distance between the palm center and each fingertip, and calculating the bending degree of fingers based on the detected distances. It then detects the bending degree of human fingers and accordingly the movement type.

In what followed next – section III – we first review the structure of the proposed model including hand detection based on skin color, detecting hand pose, palm, fingertips and fingers' bend angle. Section IV covers implementation, analysis, and comparison and discussion parts.

III. The Proposed Model

In the proposed model, for lifting objects by a robot, the user bends its fingers virtually and the robot imitates the actions. The steps for calculating fingers bending angles have been illustrated in Figure III. The model preprocesses and applies a skin filter on the input 2D image. The employed segmentation method based on skin filter can extract the hand pose from an image, even if the background contains skin-colored objects. It focuses only on the extracted areas of interest in the image for faster and more accurate analysis. In the segmented image, the model first detects the fingertips and palm, and then calculates the distance between the center of the palm and each fingertip. The bending angles of fingers have been calculated based on the distances and the model detects fingers, palm, and angles with no error. The experiments have been conducted using a normal quality webcam for capturing the user's hand movements. The user does not require to have marked gloves, wearable sensors, or even a long sleeve, and there is no limitation for the camera angle.

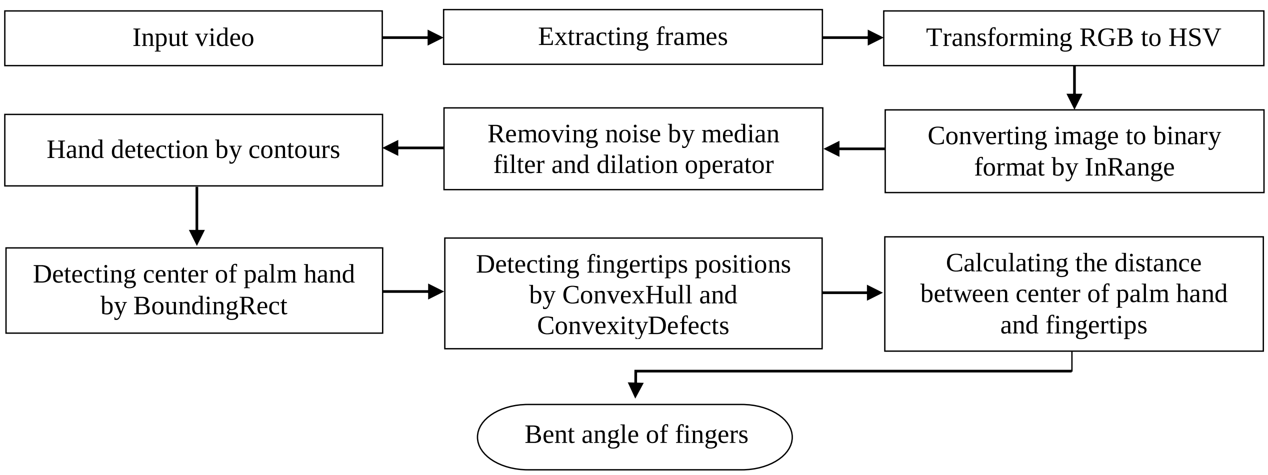

Figure III. Structure of the proposed model

A. Skin-color-based hand recognition

The most famous segmentation method in HGR systems is skin color, as it is invariant to the changes like size, rotation, and movements. The captured webcam images are in RGB and the involved parameters in this format are highly correlated and sensitive to illumination. However, HSV separates the color and illumination information. Therefore, the captured RGB images have been first converted to the HSV system and then segmented to generate the image's binary version. Each pixel of a binary image is stored in one bit, and due to the skin-colored pixels in the background, these types of images normally carry some noise. This noise creates unwanted spots in the image, as shown in Figure IV. A median filter has been used to remove isolated spots and a dilation operator for region filling.

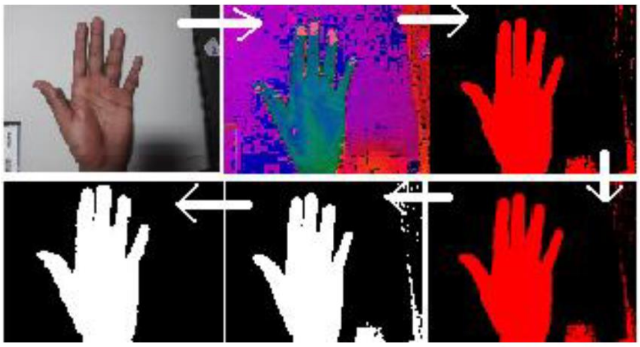

Figure IV. An example of producing a binary image: A) original images, B) HSV conversion, C) filtered image, D) smooth image, E) binary image, and F) noise-free image

B. Hand gesture recognition

The proposed model supports freedom of angle for the camera and user, and the only limitation for accurate control of the robotic hand is to face the palm hand to the camera. Contours have been used to extract the hand image from the binary image. A contour is a list of the points that represent a curve in the image and its main application is in the analysis and detection of shapes. FindContours is an OpenCV function that is used to find the largest contour in the binary image – hand shape.

C. Palm hand recognition

Finding the center of the contour helps in finding the center of the palm hand. To this end, using the BoundingRect function in OpenCV, a bounding box was applied to it and calculated the palm hand center.

D. Fingertips recognition

The ConvexHull algorithm has been employed to identify fingertips. It returns a set of polygons in which the corners of the largest one represents the fingertips. To automate the process, ConvexityDefects function approximates the gaps between the contour and the polygon by straight lines. The output of this function is multiple records of four fields: A) starting defect point, B) ending defect point, C) middle (farthest) defect point that connects starting and ending points, and D) approximate distance to the farthest point. Each record results in two lines: a line from starting point to middle point, and a line from middle point to the end point. However, the function may return more points than the number of the fingertips. Below are the steps for filtering the detected points and finding the correct location of fingertips:

- Calculating the internal angle between two defect areas in a certain period
- Calculating the angle between the starting point and contour center in a certain period
- Measuring the line length to control not exceeding a certain threshold

To find the angle between fingers, Equations 1, 2, and 3 calculate the length of the produced vectors from starting, middle, and end points. Equation 4 computes the angle.

$$p_{12} = \left(\sqrt{}(p_1.x - p_2.x)^2 + \sqrt{}\,(p_1.y - p_2.y)^2 \right) \qquad (1)$$
$$p_{13} = \left(\sqrt{}(p_1.x - p_3.x)^2 + \sqrt{}\,(p_1.y - p_3.y)^2 \right) \qquad (2)$$
$$p_{23} = \left(\sqrt{}(p_2.x - p_3.x)^2 + \sqrt{}\,(p_2.y - p_3.y)^2 \right) \qquad (3)$$
$$\theta = \arccos((p_{12}^2 + p_{13}^2 - p_{23}^2) / (2 * p_{12} * p_{13})) \qquad (4)$$

Calculating the angle between the first point of defect area and the center of the contour is an essential step for removing the non-fingertips areas. The consequent step is to filter the points to the ones between -30 to +160 degrees. Equation 5 returns the degree between the first point of defect area and center of the contour, and Equation 6 calculates the Euclidean distance – the vector length – between first to middle points. The whole process from finding the hand contour until detecting palm hand and fingertips is illustrated in Figure V.

$$\theta = \arctan\,(center.y - p2.y.center.x - p2.x) \tag{5}$$

$$length = \sqrt{}(p_2.x - p_1.x)^2 + \sqrt{}\,(p_2.y - p_1.y)^2 \tag{6}$$

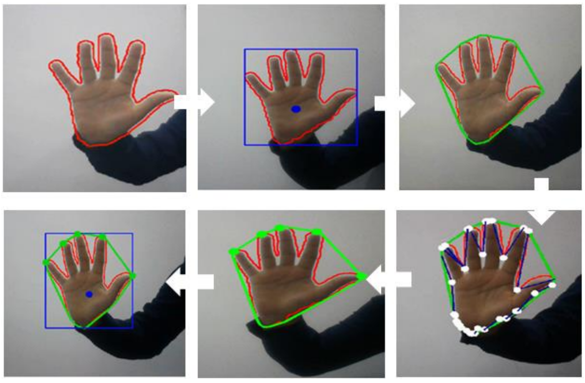

Figure V. A) hand contour, B) palm hand, C) ConvexHull polygon, D) ConvexityDefects, E) fingertips, F) palm hand and fingertips

E. Fingers' angle prediction

This research uses a combination of MLP and genetic algorithm for predicting the bent angle of fingers. MLP has a flexible structure in which neurons are located in hidden layers and each neuron can influence its input and produce the desired output. To this end, a weight applies to the input value of each neuron and the result passes into an activation function together with a bias value. Reducing classification errors and correct prediction of the pose in an artificial neural network depends on selecting optimum weight and bias values. The structure of the proposed model is shown in Figure VI, and Figure VII depicts the steps and flowchart of predicting the bent angle of fingers.

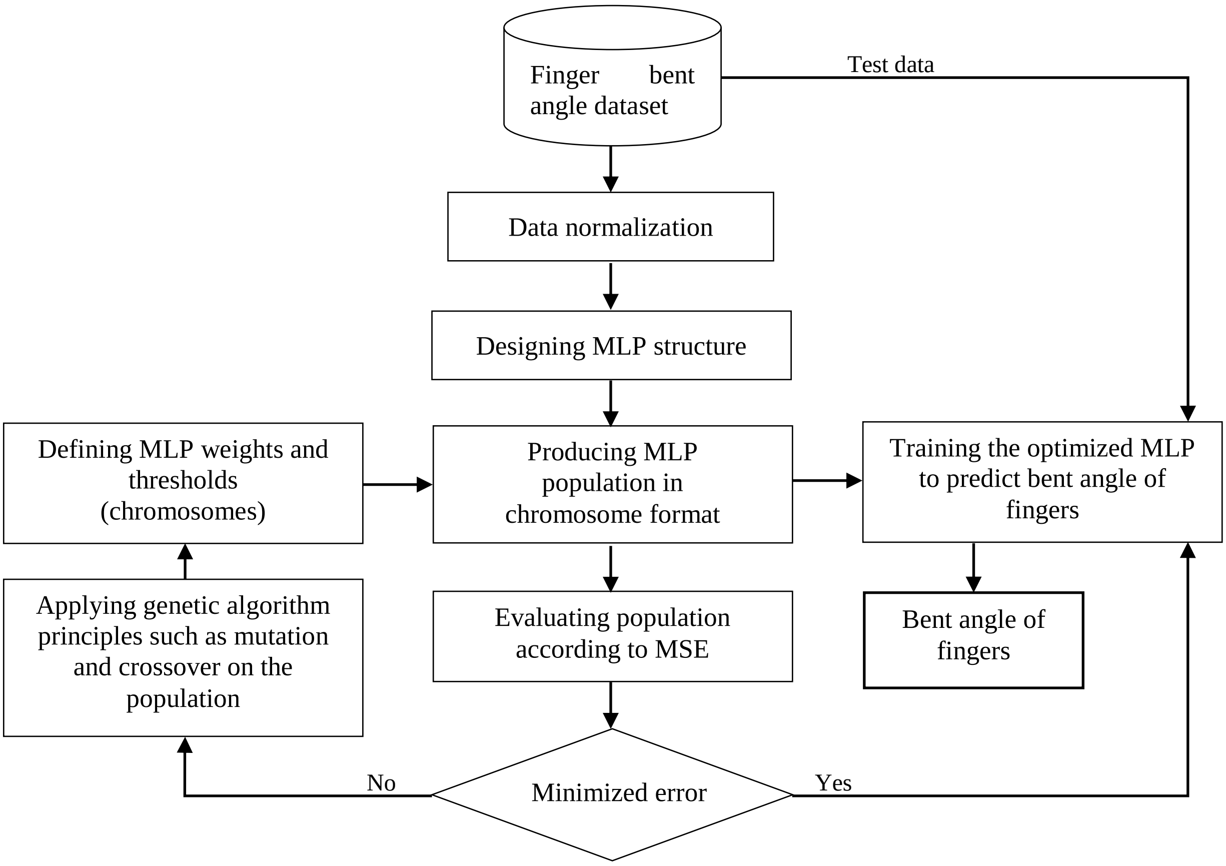

Figure VI. Model structure

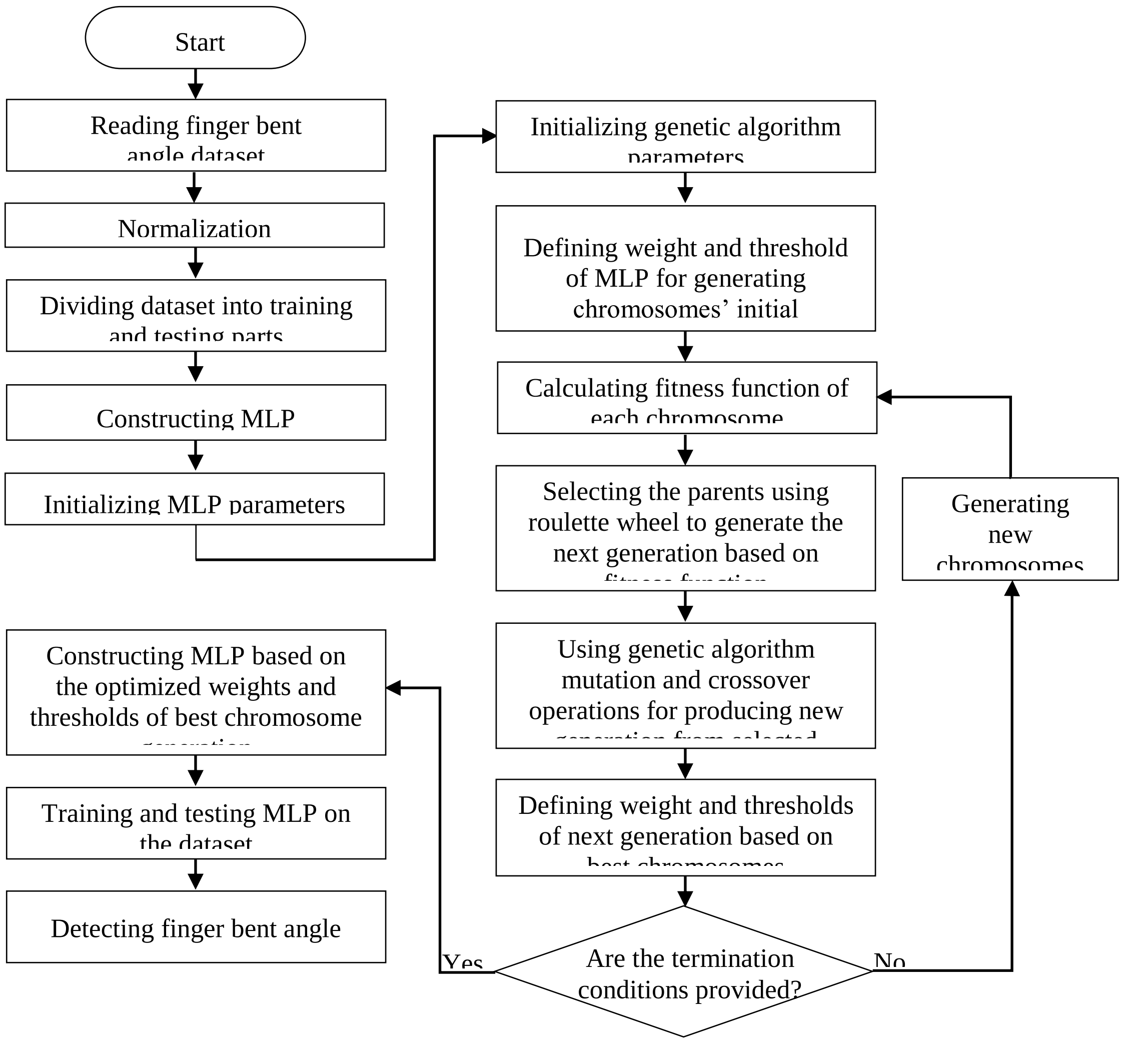

Figure VII. Flowchart of the proposed model for finger bent angle prediction

In the proposed model, the structure of each chromosome is its weight and bias in a neural network. This is formulated in Equation 7 and aims to find the optimum chromosome value using the genetic algorithm.

$$[w_{1,1},\dots w_{3,4},\dots w_{1,0},\dots w_{4,0},\dots,b_1,b_2] \qquad (7)$$

The threshold values of hidden and output layers have been denoted as b1 and b2, respectively. Upon defining the encoding system and the method of converting each answer to a chromosome, the next step is to produce the initial chromosome population. Normally, generating the initial population is a random process, however, heuristic algorithms can accelerate and optimize it. The proposed model uses a roulette wheel mechanism for selecting parents in mutation and crossover processes. In this mechanism, the probability of selecting a chromosome depends on how suitable it is for the evaluation function. In other words, the higher quality chromosome, the higher chance to be selected for producing the next generation, and vice versa. Equation 8 shows the chance of selecting a chromosome in the roulette wheel.

$$\frac{fi}{\sum_{j=1}^{n} fj} \qquad (8)$$

In the above equation, the probability of selecting the *i^th^* chromosome is the proportion of the evaluation function value of chromosome i to the sum of fitness function values of all chromosomes. Single-point crossover has been used to apply crossover in the model. This method chooses a random point in the chromosomes and swaps the information in the remaining parts. Following steps have been taken to apply single-point crossover:

- Choosing a random chromosome value between 0 and 1

- Go to the next step and mutate if the number is bigger than the mutation threshold (a value between 0 and 1), otherwise skip mutation
- Choosing a random number that indicates one of the chromosome genes and makes a numerical mutation

IV. Analysis

This research uses OpenCV functions and C++ for preprocessing, segmentation, and feature extraction from video. The functions help to detect the center of the palm hand, fingertips, and bent angle of fingers. Matlab is the chosen platform due to having rich programming interface and metaheuristic libraries. Applying genetic algorithm principles on MLP helps the model to enhance predicting fingers' bent angle. Followings are the steps taken for implementing the proposed model in Matlab:

- Defining the initial parameters such as population, number of iterations, mutation rate, and crossover rate of genetic algorithm
- Preprocessing and normalizing part of input data to reduce learning errors and enhance performance in hand movement detection
- Dividing the dataset into training and evaluation parts
- Producing chromosomes for MLP and applying mutation and crossover on the chromosomes to find optimum weight and bias and reduce the error rate
- Evaluating the model according to the metrics in section 4.b.

A. Dataset

This research has been implemented by a data mining and machine learning approach. In the first step, the data for training the genetic-based MLP produced using image processing and hand pose analysis techniques. To this end, hand pose images of more than 300 people were analyzed and the required features were extracted. Analysis of each image produced a record consisting of information of the distances between fingertips to the center of the palm, and the hand pose.

B. Evaluation metrics

In the proposed model, each hand pose has been considered as a class and is defined by a set of features. Therefore, this research challenges a multiclass optimization problem. The proposed model classifies each given sample in its corresponding class and tries to minimize the average learning and classification errors. The mean square error (MSE) and root mean square error (RMSE) have been calculated by Equation 9 and 10 as goal test measures. In these equations, *n* represents the number of the samples, and real and predicted values of an instance – such as *I* – are shown by o_i and $\widehat{o_i}$, respectively.

$$MSE = \frac{\sum_{i=1}^{n}(\widehat{o_i} - o_i)^2}{n} \qquad (9)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\widehat{o_i} - o_i)^2}{n}} \qquad (10)$$

Two other important parameters for evaluating the applicability of an algorithm include sensitivity and accuracy. Sensitivity is the proportion of true positive detections to the sum of true positives and false negatives – Equation 11 – and accuracy is the ratio of truly detected positives and negatives to all detections – Equation 12.

$$Sensitivity = \frac{TP}{TP + FN} \qquad (11)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (12)$$

C. Error analysis by increasing population and iterations

The experiments were conducted by two populations of 5 and 10. The number of iterations is set to 20, and mutation and crossover ratios are 0.15 and 0.5, respectively. A chromosome is a two-layer MLP in which each layer has four neurons. MSE prediction errors of both populations have been reported in Figures VIII and IX.

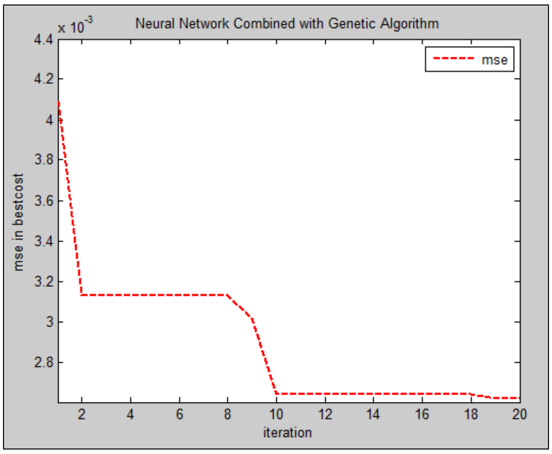

Figure VIII. Prediction error rate with 5 chromosomes in the proposed model

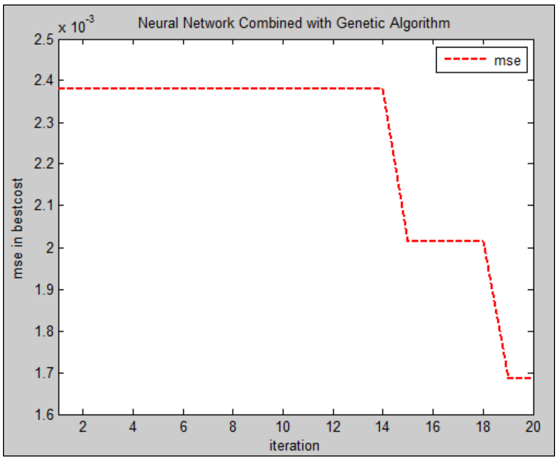

Figure IX. Prediction error rate with 10 chromosomes in the proposed model

The reported values in Figures VIII and IX prove that increasing the number of chromosomes results in reducing MSE for hand pose detection. It happens due to the following reasons:

- Increasing the number of chromosomes is increasing the number of neural networks for a prediction that results in a more accurate classification
- Increasing population produces more diverse test ratios in the genetic algorithm and accordingly increases the chance of finding a final and accurate answer
- Increasing chromosomes in the genetic algorithm increases problem search space that normally results in optimum answers and error reduction
- Increasing population increases the number of elite members, enhances the probability of mutation and crossover between this population, and leads to increasing the chance of having a more accurate MLP for hand pose prediction

In addition to the population, the number of iterations can help reduce hand pose detection errors. Behavioral analysis of Figures VIII and IX reveals that MSE and the number of iterations have a reverse correlation. In other words, increasing iterations gives the chromosomes more chance to choose optimum ratios in the MLP, which leads to lower MSE.

D. Comparison and discussion

To analyze and compare the proposed genetic-based MLP model has been compared with a standard MLP. Both implementations have four hidden neurons, a sigmoid activation function, 15 populations, 30 iterations, 0.15 mutation, and 0.5 crossover, and they dedicate 75% of the dataset to training and 25% to testing processes. Figure X demonstrates the MSE difference between the proposed model and the standard MLP, and Figure XI shows the comparison according to the RMSE benchmark.

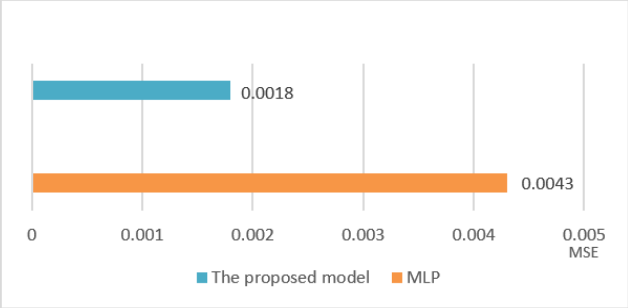

Figure X: Comparing MSE of the proposed model and MLP

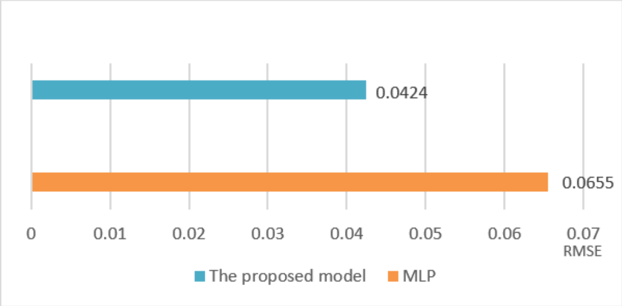

Figure XI. Comparing the RMSE of the proposed model and MLP

The experiments and analysis of MSE and RMSE results demonstrate that the proposed model is more accurate than MLP. The index values in the proposed model are 0.0018 and 0.0424, while for MLP, they increase to 0.0043 and 0.0655, respectively. In other words, applying the genetic algorithm in the proposed model reduces MSE by 58.13% and RMSE by 35.26%. These lower error rates prove that the genetic algorithm decreases the hand-pose detection errors and boosts the learning process.

50 experiments were conducted according to the configurations above to compare the proposed model with MLP in terms of sensitivity and accuracy. As shown in Figure XII, the sensitivity of the proposed model and MLP is 93.42% and 83.17%, respectively. In terms of accuracy, the proposed model has a 6.65% higher performance—92.27% to 85.62%. These results prove that the proposed combination of genetic algorithm and MLP delivers more reliable results for analyzing human hand movements.

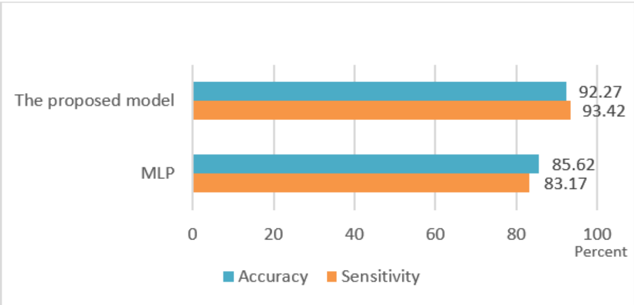

Figure XII. Comparing sensitivity and accuracy of the proposed model and MLP

V. Conclusion

Training robots help humans to utilize them in dangerous environments, remote operations, or even industry. However, a major problem in this area is choosing an efficient learning algorithm and conducting the correct training process. MLP is one of the most common supervised learning techniques that play a key role in robot learning. This research proposes a novel model for training and controlling robots based on applied genetic algorithm principles on MLP. It uses a set of hand pose features based on the distances between fingertips and the center of the hand palm. Genetic algorithm principles such as concentrating on elite populations, mutation, and crossover have been used to reduce error rates. The experiments show less 58.13% MSE and 35.26% RMSE errors in hand pose detection compared to standard MLP. In terms of sensitivity and accuracy, the proposed model reports values of 93.42% and 92.27%, while for MLP, these are 83.17% and 85.62%, respectively.

References

[1] Yazdanpanah S, Chaeikar SS, Jolfaei A. Monitoring the security of audio biomedical signals communications in wearable IoT healthcare. Digital Communications and Networks. 2022.

[2] Chaeikar SS, Jolfaei A, Mohammad N, Ostovari P. Security Principles and Challenges in Electronic Voting. In2021 IEEE 25th International Enterprise Distributed Object Computing Workshop (EDOCW) 2021 Oct 25 (pp. 38-45). IEEE.

[3] Chaeikar SS, Manaf AA, Alarood AA, Zamani M. PFW: Polygonal Fuzzy Weighted—An SVM Kernel for the Classification of Overlapping Data Groups. Electronics. 2020 Apr;9(4):615.

[4] Chaeikar SS, Jolfaei A, Mohammad N. AI-Enabled Cryptographic Key Management Model for Secure Communications in the Internet of Vehicles. IEEE Transactions on Intelligent Transportation Systems. 2022 Aug 29.

[5] Lin CS, Chen PC, Pan YC, Chang CM, Huang KL. The Manipulation of Real-Time Kinect-Based Robotic Arm Using Double-Hand Gestures. Journal of Sensors. 2020 Jan 25;2020.

[6] Chaeikar SS, Zamani M, Manaf AB, Zeki AM. PSW statistical LSB image steganalysis. Multimedia Tools and Applications. 2018 Jan;77(1):805-35.

[7] Shojae Chaeikar S, Tadayon M H, Jolfaei A, Alizadeh M. An intelligent cryptographic key management model for secure communications in distributed industrial intelligent systems. International Journal of Intelligent Systems.

[8] Yazdanpanah S, Shojae Chaeikar S. Secure SMS Method Based on Social Networks. International Journal of Scientific Research in Science, Engineering and Technology. 2016: 2(6): 368-376.

[9] Yazdanpanah S, Shojae Chaeikar S. IKM-based Security Usability Enhancement Model. IRACST-International Journal of Computer Science and Information Technology & Security (IJCSITS). 2012 Aug(4).

[10] Chaeikar SS, Ahmadi A. Ensemble SW image steganalysis: A low dimension method for LSBR detection. Signal Processing: Image Communication. 2019 Feb 1;70:233-45.

[11] Chaeikar SS, Ahmadi A. SW: A blind LSBR image steganalysis technique. InProceedings of the 10th International Conference on Computer Modeling and Simulation 2018 Jan 8 (pp. 14-18).

[12] Ramya R, Srinivasan K. Real time palm and finger detection for gesture recognition using convolution neural network. In Human Behaviour Analysis Using Intelligent Systems 2020 (pp. 1-19). Springer, Cham.

[13] Chaeikar SS. *Pixel Similarity Weight for Statistical Image Steganalysis* (Doctoral dissertation, Universiti Teknologi Malaysia).

[14] Yazdanpanah S, Shojae Chaeikar S, Zamani M, Kourdi R. Security features comparison of master key and IKM cryptographic key management for researchers and developers. In International Conference on Software Technology and Engineering, 3rd(ICSTE 2011) 2011. ASME Press.

[15] Shojae Chaeikar S, Jafari M, Taherdoost H, Kar NS. Definitions and criteria of CIA security triangle in electronic voting system. International Journal of Advanced Computer Science and Information Technology. 2012 Oct;1(1):14-24.

[16] Karamizadeh, Sasan, Saman Shojae Chaeikar, and Alireza Jolfaei. "Adult content image recognition by Boltzmann machine limited and deep learning." *Evolutionary Intelligence* 16.4 (2023): 1185-1194.

[17] Chaeikar, S.S., Yazdanpanah, S. and Chaeikar, N.S., 2021. Secure SMS transmission based on social network messages. *International Journal of Internet Technology and Secured Transactions*, *11*(2), pp.176-192.

[18] Honarbakhsh S, Masrom M, Zamani M, Chaeikar SS, Honarbakhsh R. A Trust Based Clustering Model for Dynamic Monitoring in Ad hoc Network. InInternational Conference on Computer and Computational Intelligence (ICCCI 2010) 2010 Dec 25.

[19] Karamizadeh, S., Moazen, M., Zamani, M., & Manaf, A. A. (2024). Enhancing IoT-Based Smart Home Security Through a Combination of Deep Learning and Self-Attention Mechanism. Arabian Journal for Science and Engineering, 1-11.

[20] Arteaga MV, Castiblanco JC, Mondragon IF, Colorado JD, Alvarado-Rojas C. EMG-driven hand model based on the classification of individual finger movements. Biomedical Signal Processing and Control. 2020 Apr 1;58:101834.

[21] Nakhaie, M. M., Karamizadeh, S., Shiri, M. E., & Badie, K. (2023). Using Convolutional Neural Network to Enhance Classification Accuracy of Cancerous Lung Masses from CT Scan Images. Journal of AI and Data Mining, 11(4), 547-559.

[22] Lupinetti K, Ranieri A, Giannini F, Monti M. 3D dynamic hand gestures recognition using the Leap Motion sensor and convolutional neural networks. InInternational Conference on Augmented Reality, Virtual Reality and Computer Graphics 2020 Sep 7 (pp. 420-439). Springer, Cham.

[23] Karamizadeh, S., & Arabsorkhi, A. (2018). Skin Classification for Adult Image Recognition Based on Combination of Gaussian and Weight-KNN. International Journal of Information and Communication Technology Research, 10(2), 56-62.

[24] Cobos-Guzman S, Verdú E, Herrera-Viedma E, Crespo RG. Fuzzy logic expert system for selecting robotic hands using kinematic parameters. Journal of Ambient Intelligence and Humanized Computing. 2020 Apr;11(4):1553-64.

[25] Karamizadeh, S., Abdullah, S. M., Shayan, J., Nooralishahi, P., & Bagherian, B. (2017). Threshold based skin color classification. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(2-3), 131-134.

[26] Martinelli, D., Sousa, A.L., Augusto, M.E., Kalempa, V.C., de Oliveira, A.S., Rohrich, R.F. and Teixeira, M.A., 2019, October. Remote control for mobile robots using gestures captured by the rgb camera and recognized by deep learning techniques. In *2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE)* (pp. 98-103). IEEE.

[27] Sagayam KM, Hemanth DJ, Vasanth XA, Henesy LE, Ho CC. Optimization of a HMM-Based Hand Gesture Recognition System Using a Hybrid Cuckoo Search Algorithm. InHybrid Metaheuristics for Image Analysis 2018 (pp. 87-114). Springer, Cham.

[28] Karamizadeh, S., & Arabsorkhi, A. (2017). Enhancement of Illumination scheme for Adult Image Recognition.Karamizadeh, S., & Arabsorkhi, A. (2017). Enhancement of Illumination scheme for Adult Image Recognition.

[29] Arab, F., Zamani, M., Karamizadeh, S., Alizadeh, M., Chaeikar, S.S. and Khodadadi, T., 2022, April. Comparison of Data Hiding Techniques for Video Watermarking Applications. In *2022 7th International Conference on Computer and Communication Systems (ICCCS)* (pp. 168-173). IEEE.
[30] Mirjalili S, Lewis A. The whale optimization algorithm. Advances in engineering software. 2016 May 1;95:51-67.
[31] Karamizadeh, S., Abdullah, S. M., Shayan, J., Zamani, M., & Nooralishahi, P. (2017). Taxonomy of filtering based illumination normalization for face recognition. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(1-5), 135-139.
[32] Mirjalili S, Mirjalili SM, Lewis A. Grey wolf optimizer. Advances in engineering software. 2014 Mar 1;69:46-61.
[33] Dhiman G, Kumar V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. Knowledge-Based Systems. 2019 Feb 1;165:169-96.
[34] Dehzangi, A., Foladizadeh, R. H., Aflaki, M., & Karamizadeh, S. (2011). The application of fusion of heterogeneous meta classifiers to enhance protein fold prediction accuracy. In Intelligent Information and Database Systems: Third International Conference, ACIIDS 2011, Daegu, Korea, April 20-22, 2011, Proceedings, Part I 3 (pp. 538-547). Springer Berlin Heidelberg.
[35] Dhiman G, Kumar V. Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. Knowledge-Based Systems. 2018 Nov 1;159:20-50.
[36] Karamizadeh, S., & Abdullah, S. M. (2018). Race classification using gaussian-based weight K-nn algorithm for face recognition. Journal of Engineering Research, 6(2), 103-121.
[37] Azarnik, A. H. M. A. D., SHAYAN, J., ZADEH, S. K., & PASHANG, A. (2013). Lightweight authentication for user access to Wireless Sensor networks. In Proceedings of the 12th WSEAS Int. Conf. on Electronics, Hardware, Wireless and Optical Communications (EHAC'13), Cambridge, UK (pp. 35-39).
[38] Dhiman G, Kumar V. Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Advances in Engineering Software. 2017 Dec 1;114:48-70.
[39] Mirjalili S, Mirjalili SM, Saremi S, Mirjalili S. Whale optimization algorithm: theory, literature review, and application in designing photonic crystal filters. Nature-Inspired Optimizers. 2020:219-38.
[40] Shadravan S, Naji HR, Bardsiri VK. The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Engineering Applications of Artificial Intelligence. 2019 Apr 1;80:20-34.
[41] Moosavi SH, Bardsiri VK. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. Engineering Applications of Artificial Intelligence. 2019 Nov 1;86:165-81.
[42] Karamizadeh, S., Abdullah, S. M., Manaf, A. A., Zamani, M., & Hooman, A. (2013). An overview of principal component analysis. Journal of signal and information processing, 4(3), 173-175.
[43] Chaeikar, S.S., Ahmadi, A., Karamizadeh, S. and Chaeikar, N.S., 2022. SIKM–a smart cryptographic key management framework. *Open Computer Science*, *12*(1), pp.17-26.
[44] Alizadeh M, Hassan WH, Zamani M, Khodadadi T, Shojae Chaeikar S. A prospective study of mobile cloud computing. International Journal of Advancements in Computing Technology. 2013;5(11):198-210.
[45] Karamizadeh, S., Abdullah, S. M., Halimi, M., Shayan, J., & javad Rajabi, M. (2014, September). Advantage and drawback of support vector machine functionality. In 2014 international conference on computer, communications, and control technology (I4CT) (pp. 63-65). IEEE.
[46] Wei H, Bu Y, Zhu Z. Robotic arm controlling based on a spiking neural circuit and synaptic plasticity. Biomedical Signal Processing and Control. 2020 Jan 1;55:101640.
[47] Mazhar O, Navarro B, Ramdani S, Passama R, Cherubini A. A real-time human-robot interaction framework with robust background invariant hand gesture detection. Robotics and Computer-Integrated Manufacturing. 2019 Dec 1;60:34-48.
[48] Luzhnica G, Simon J, Lex E, Pammer V. A sliding window approach to natural hand gesture recognition using a custom data glove. In2016 IEEE Symposium on 3D User Interfaces (3DUI) 2016 Mar 19 (pp. 81-90). IEEE.
[49] Hung CH, Bai YW, Wu HY. Home outlet and LED array lamp controlled by a smartphone with a hand gesture recognition. In2016 IEEE international conference on consumer electronics (ICCE) 2016 Jan 7 (pp. 5-6). IEEE.
[50] Chen Y, Ding Z, Chen YL, Wu X. Rapid recognition of dynamic hand gestures using leap motion. In2015 IEEE International Conference on Information and Automation 2015 Aug 8 (pp. 1419-1424). IEEE.
[51] Haria A, Subramanian A, Asokkumar N, Poddar S, Nayak JS. Hand gesture recognition for human computer interaction. Procedia computer science. 2017 Jan 1;115:367-74.
[52] Chaudhary A, Raheja JL. Bent fingers' angle calculation using supervised ANN to control electro-mechanical robotic hand. Computers & Electrical Engineering. 2013 Feb 1;39(2):560-70.
[53] Marium A, Rao D, Crasta DR, Acharya K, D'Souza R. Hand gesture recognition using webcam. American Journal of Intelligent Systems. 2017;7(3):90-4.
[54] Simion G, David C, Gui V, Căleanu CD. Fingertip-based real time tracking and gesture recognition for natural user interfaces. Acta Polytechnica Hungarica. 2016 Jan 1;13(5):189-204.