International Journal of Applied Engineering Research ISSN 0973-4562 Volume 19, Number 1 (2024) pp. 41-49
 https://dx.doi.org/10.37622/IJAER/19.1.2024.41-49

Human Hand Gesture Recognition Based Feature Extraction and Classification Using Sign Language Data

[1]R. Siji, and [2]Dr. S. Nithya

[1]Research Scholar, Department of Computer Science, Park's College Tirupur, Email id- sijiramachandran@gmail.com

[2]Assistant professor, Dept of Computer Science, A.V.P College of Arts and Science Tirupur, Email id- drnithyasundaram23@gmail.com

Abstract

Hand gestures are natural forms of Human-Computer Interaction (HCI).Hand gesture recognitions are rapidly developing fields of study in computer vision and machine learning as they offer wide ranges of potential applications, allowing people to interact with robots and system interfaces in more straightforward and natural ways without requirements for additional hardware. Hence, gesture recognition studies aim to develop systems that can recognize certain human gestures and utilize them to operate devices or transmit information. In order to accompolish real-time gesture recognitions, reliable hand detections are necessary for vision-based hand gesture interfaces. The initial part of this work is data collection, and standard dataset is introduced for data validations. Dataset with 26 signs was created using Sign Language (SL) and assessed in real time for a variety of lighting scenarios. The suggested technique recognises motions and identifies human hands using a feature set of symmetric patterns that is obtained using techniques for extracting hand perimeter features. Depth based Speeded-Up Robust Features (DSURF) is employed to identify important focal spots in SL-based images along with their associated characteristics. Finally, hand gestures are classified using Improved Graph Neural Networks (IGNN) trained on symmetric patterns under varying illumination conditions. Selected characteristics are validated where experimental results suggested that this work's schema outperformed other methods in twerms of precision for a variety of illumination conditions and ranging pixel sizes.

Keywords: Sign Language; gesture recognition; variant lighting conditions; symmetric pattern; accuracy

INTRODUCTION

Hand gestures are body languages expressed using hand palm centres, finger placements, and hand formations. The gestures fall into two categories: dynamic gestures are made up of a sequence of hand motions, such as waving, and static gestures are stable hand forms. Hand gestures differ like handshakes dependent on individuals, locations and times. The primary distinction between gestures and postures are that the former emphasises hand movements, while the latter stresses on hand shapes. The two basic techniques used in studies for hand gestures the use of sensors on wearable gloves and camera visions [1, 2].

Hand gestures are natural ways to connect and assist communication in a variety of contexts for researches. Previously, wearable sensors were directly affixed to hands with gloves and used for recognizing hand gestures. These sensors were able to identify bodily reactions based on hand gestures or finger bending. Computers attached to wire-connected gloves subsequently processed the gathered data. By utilising a sensor coupled with a microprocessor, this glove-based sensor system might be made portable.

The development of hand gestures for HCI began with e developments of glove sensor data shown in Figure 1. They provided basic computer interface instructions. By determining precise coordinates of palms and fingers' locations, gloves employed multiple sensor types to record hand motions and positions [3]. These sensors types included curvature [4], angular displacement [5], optical fibre transducer [6], flex [7], and accelerometer [8] sensors and were based on bending angles. The sensors made use of various principles.

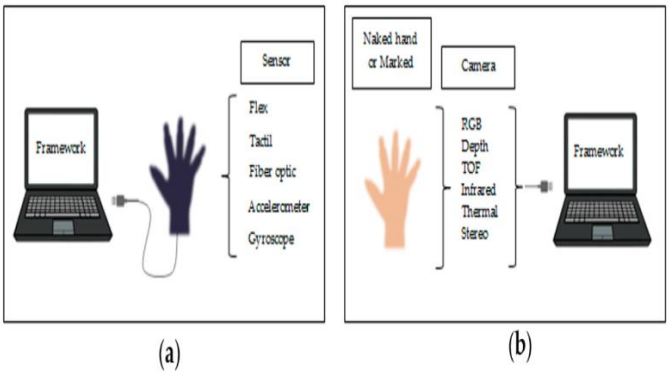


Figure 1: Various methods for hand gestures. (a) Glove based sensors connected computer's or portable devices; (b) Photographic vision of marked gloves or naked hands

Recognition systems become less accurate when there are image noises. Unwanted information that is present with input images is referred to as noise. For instance, changes in the brightness levels of individual images. This fluctuation in lighting can be attributed to either the image sensor or the surrounding image. Brightness variation is the term used to describe the fluctuation in the lumens (lm) that the lighting equipment emits. Greater lumen counts indicate stronger light, lower lumen counts indicate faint lights. Bedrooms and baths have lower illumination, whereas rooms like kitchens are often set to general lighting. When users wander around the house under varying lighting conditions and with unanticipated fluctuations in brightness, real-time recognitions become more challenging. Image improvements are therefore required for accurate recognitions. In the last ten years, a number of research using computer vision methods have been published. Murthy et al. [9] described limits of computer vision under different settings and highlighted basic HCI techniques for recognitions and categorizations. Khan et al. [10] proposed recognitions using feature extractions and gesture classifications. Suriya et al. [11] detailed about hand gesture detection techniques and algorithms for mouse control applications in HCI. They also provided outlines of hidden Markov models (HMM).Sonkusare et al.'s study [12] examined many approaches and compared them based on tracking, feature extraction, hand segmentation methodology, and recognition techniques. It came to the conclusion that the rate of recognition was a trade-off, with the temporal rate being constrained by processing capacity. Lastly, Kaur et al. [13] examined approaches for hand gesture identifications based on sensors and visions in an effort to integrate existing techniques and increase algorithmic precisions.

Various image enhancement approaches have been presented by numerous researchers to increase the process of gesture recognition and feature extraction. A high degree of accuracy and dependability is required for some applications, including sign language identification for recognitions of movements. For this reason, it is necessary to identify distinctive qualities. However, there is still much room for the development in the field of gesture identification in many lighting conditions, as the great majority of efforts have just examined one lighting environment to recognise the gesture pattern. This study looks at how symmetric patterns and a related luminosity-based filter may be used to identify gestures. The following are the key contributions of this paper:

Initial proposal includes gesture detection frameworks based on symmetry patterns that perform in a variety of lighting conditions.

Secondly, 26 SL hand gesture images taken in various lighting scenarios generate a dataset and an effective technique based on perimeter feature extractions and luminosity-based grey-scale image conversions are applied to extract gesture features.

Thirdly, process of segmenting and identifying significant points reduces time required for key point localizations inside features and increases DSURF key point counts.

In order to improve recognition accuracy and prevent uncertainty management during decision-making, IGNN is used to verify gesture recognitions.

Finally, comparison of this work with previously published information on related fields are executed to show efficiency ofthe suggested framework.

The following parts comprise the paper's organisation: Section 2 is a review of the literature. Section 3 describes the suggested hand gesture recognition structure using mathematical modelling in detail. The testbed environment and outcomes were presented in Section 4. Section 5 concluded with discussing the conclusion and potential future work.

RELATED WORKS

Zuocai Wang et al. (2018) [14] proposed particle filtering to identify hand gestures. This filtering technique was used on images of hand gestures with similar backdrops. Their results showed 90.6% accuracy, 84.7% specificity, and 92.1% sensitivity.

Suguna and Neethu (2017) [15] by identifying form features from the photos, we were able to categorise hand gesture photographs into numerous groups. These collected attributes were trained for classifications using k-means clustering. Marium et al. (2017) [16] suggested recognitions of hand gestures based on convexity algorithms and their filters on hand gestures used same image backgrounds. The authors' results showed 87.5% accuracy, 82.1% specificity, and 90.7% sensitivity. This method's primary drawback was that it worked best with static backgrounds for hand gesture images.

The work in [17] proposed compensations for poor ambient illuminations by balancing them against incident illuminations. The study in [18] detailed about information on imaging technologies, process of collecting data, structures of databases, many possible applications for databases, and means of obtaining databases. The study gathered more than 40,000 images of 68 individuals for its database. Every individual is imagegraphed in four distinct emotions, in thirteen distinct stances, and in forty-three different lighting scenarios. A large-scale dataset with different light fluctuations were compiled in [19] for assessing effectiveness of Remote Image Plethysmography (RPG) techniques. They also suggested low-light enhancements for remote heart rate assessments in dimly lit environments.

The study in [20] proposed fine perceptive generative adversarial networks (FP-GANs) for generating super-resolution (SR) magnetic resonance (MR) images from their low-resolutions. FP-GANs used divide-and-conquer techniques for handling high/low-frequency components of MR images independently and concurrently. Tensored GANs with high-order pools were employed in [21] to identify Alzheimer's disease and mild cognitive impairments. The proposed method tensorized 3 player cooperative game frameworks for utilizing brain architectural advantages. The inclusion of high-order pools in classifiers were suggested to obtain2nd order statistics in MR images and understand outcomes of machine learning models by mapping and integrating synergic muscle activities and hand motions were detected.The study [22] proposed cutting-edge XAI algorithms for categorising EMG hand gestures.

METHODOLOGY

Framework for sensor-based sign language gesture detections is suggested and detailed in this section. It comprises of four primary phases: name image acquisitions, image pre-processes, feature extractions, and classifications and depicted as Figure 1,

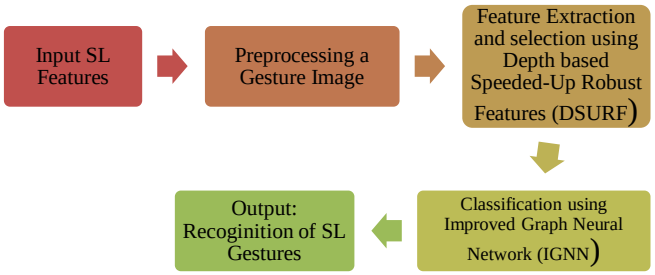


Figure 1: Proposed methodology Flow Diagram

A. Input data Pre-processing

The first step in the identification procedure is to use the Kinect sensor to capture deep images. The distance between an RGB images object and image plane is reflected in each pixel of a deep image. After that, hand forms are accurately split to detect and track hand shapes in a manner like to human communication. SL-based gesture images are transformed into PNG format and kept in a database in order to create the datasets. The subsequent stage involves pre-processes used on obtained images to convert them into uniform brightness levels. In order to do it, the system converts the input image to greyscale and applies the luminosity approach. Image segmentation difficulties are among the many activities that benefit greatly from grayscale conversion, which also makes it much easier to deal with. The weighted approach, often referred to as the luminosity method, is used to convert an image to grayscale. Luminosities were primarily proposed to equalise weights of R, G, and B values based on their wavelengths in images. An improved variation of the average approach is the luminosity method. The following Equation can be used to determine luminosity-based greyscale conversion, as covered in [23]:

$$Grayscale = 0.299R + 0.587G + 0.114B (1)$$

Here Red, Green, and Blue are represented by the letters R, G, and B, correspondingly.

B. DSURF for feature extraction and selection

Extracting the relevant traits and choosing them is the next step. Choosing how many features to include is essential as more features need more storage and processing power. The accuracy is impacted by fewer features. To identify and extract the relevant characteristics from gathered data, the DSURF approach is provided. Features including perimeters, hand sizes, hand centres and finger distances are extracted from input images.

Similar in concept to SIFT, SURF was created by Bay et al. [24] as a more reliable and speedier extractor than earlier models. In contrast to SIFT, it made use of integral images [25] and Fast-Hessian detectors based on 2D Haar wavelet responses generated simpler filtering kernels. Their descriptors comprised of local gradient information around keypoints which included 2D Haar wavelet responses to local regions, SIFT, and gradient estimating windows.

The keypoint detection equations using a Gaussian kernel are as follows.

$$L(x,\sigma) = G(x,\sigma) * I(x)$$

$$DoG(x,k^m\sigma_s) = L(x,k^{m+1}\sigma_s) - L(x,k^m\sigma_s)$$

where $x = (x,y)$; x and y are indices of columns and rows, I represents input images, G implies Gaussian functions with standard deviations σ,$*$ signifies convolution operators, DoG (difference of Gaussian) values were computed as differences of Ls is calculated by k^m and $k^{(m+1)}$, here s implies octave scales, σ_s stands forstandard deviations in s-th octaves, k implies constants, and m represents indices of s-th octaves. Keypoint computations of local extremas of DoGs were based on deviations in standard deviation values. Octaves had keypoints and differently scaled keypoints were located by adjusting previous octaves twice i.e., $\sigma_s = 2\sigma_{s-1}$. Multi-octave method is helpful for locating key points of different dimensions on 2D images (Refer Figure 2). SURF is a descriptor and detector of local features that may be applied to various applications including 3D reconstructions, object identifications, registrations, and classifications. They are scales and in-plane rotation invariants. When detectors discover interesting points in images, descriptors construct feature vectors these points and detail their characteristics. SURF characteristics are mainly insensitive to lighting and affine modification, as well as shifting, rotation, and scaling.

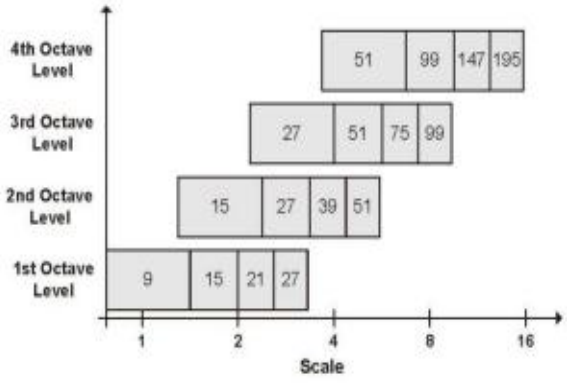


Figure. 2 Box filter sizes for different octave levels in SURF algorithm

Classical SURF algorithms aspply Hessian matrices as shown below:

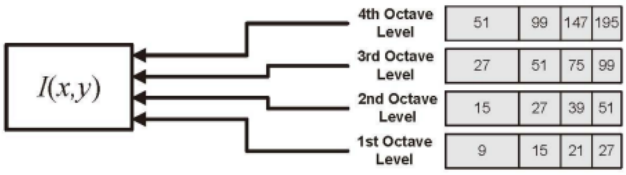
$$H(x,\sigma) = \begin{pmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{xy}(X,\sigma) & L_{yy}(X,\sigma) \end{pmatrix} (2)$$

Where $L_{xx}(X,\sigma)$ represents Gaussian 2nd order derivative convolutions $\frac{\partial^2}{\partial x^2} g(\sigma)$ with hand images I at points x, and similarly for $L_{xy}(X,\sigma)$and $L_{yy}(X,\sigma)$.

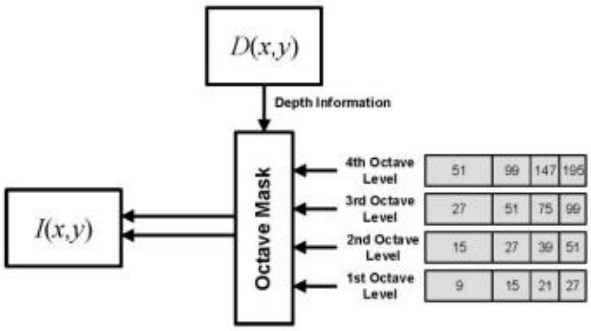
Larger size feature points are often taken from objects that are closer to deeper parts. A large-scale keypoint is produced when a high octave level is employed. Keeping this in mind, the keypoints may be generated based on the distance to the keypoints by integrating depth data. This methodology is used to suggest a keypoint detection method, as shown in Fig. 3(b), instead of the standard SURF algorithm, as presented in Fig. 3(a), which utilises an octave mask dependent on depth data. The suggested approach is shown as follows.

$$DoG_M(x, k^m\sigma_s, M^s(x)) = \begin{cases} L(x, k^{m+1}\sigma_s) - L(x, k^m\sigma_s), if\ M^s(x) = 1 \\ 0, if\ M^s(x) = 0 \end{cases} \quad (3)$$

Where $M^s(x)$ represents sth octave masks with values of 0 or 1. If $M^s(x) = 0$, DoG processes are skipped for reducing computations.



(a) Classical SURF algorithm



(b) DH-SURF algorithm

Figure. 3 Overview of applying box filters

For generating $M^s(x)$, Deep data handling is a priority. As seen by the dark areas in Fig. 4(a), the initial depth data included unmeasured data due to the sensor system. Consequently, as seen in Fig. 4(b), image inpaints (Telea, 2004) filled depth data that were not measured. One common method for repairing minor damage to an image is image inpainting. Using known neighbourhood data and a normalised weighting function, the method fills in the gaps in the depth data. By employing the FMM (Fast Marching Method) outlined in Sethian (1996), the algorithm is incredibly easy to construct and executes quickly. Following the inpainting of unmeasured depth data, the octave masks are produced in the following manner based on distance information:

$$M^s(x) = \begin{cases} 1, if\ d_s - \sigma_{d_s} \le D(x) \le d_{s-1} + \sigma_{d_{s-1}} \\ 0, otherwise \end{cases} \quad (4)$$

Where $D(x)$ returns pixelwise depth data of $x, d_s = 2 \cdot d_{s-1}$, where ds are conditional distances of octave levels, and σ_{d_s}represents standard deviations of depth values at d_s of sensors. Higher octave mask values near 1 in close proximity, while lower octave masks are set to 1. Ambiguities caused by sensor noises are avoided using standard deviations ds which produce marginal gaps that overlap octave masks at bordering points, d_s, of octave masks. The depth-based SURF (DSURF) is named as depth information is included into SURF information. The DSURF extraction procedure utilising the masks created from the depth value is displayed in Fig. 5.The various octave masks are used to determine which characteristics are taken from each octave. DSURF is executed by integrating characteristics from several octaves. The scale-invariance feature is maintained despite fewer octaves and duplicated keypoints are present thanks to the depth data. Moreover, fewer keypoints mean a shorter processing time.

The perimeter feature extraction (PFE) approach detects borders of human hand by obtaining picels counts with 1 and 0 values for neighbouring picture elements whereas bypassing grey shade values resulting in perimeter value computations of hand's perimeter. The hand's projection, which yields the hand's size, is found in order to compute the form. In order to do it, the entire rows and column values are added together to determine the vertical and horizontal values as follows:

$$v_i(c) = \sum_{c=0}^{n-1} P_i(r, c) \quad (5)$$

$$h_i(r) = \sum_{r=0}^{n-1} P_i(r, c) \quad (6)$$

Where, for pixels P in images, h and v stand for horizontal and vertical locations, and r and c for rows and columns. Hand's vertical side or greatest "height" values are represented by "n" in Equation (5) while horizontal side's i.e. greatest "width" are denoted by "n" in Equation (6). Given that hand sizes vary from person to person, the hand size feature is helpful in identifying changes in hand size. A hand's size component depicts the size of the hand at a certain moment in time. The function, $M_i(r, c)$ is defined as Equations (7) and (8) to compute hand sizes.

$$M_i(r, c) = \begin{cases} 1\ if\ P(r, c) = & ith\ object\ number \\ 0 & otherwise \end{cases} \quad (7)$$

$$A_i = \sum_{r=0}^{n-1} \sum_{c=1}^{n-1} M_i(r, c) \quad (8)$$

Where r and c imply rows and columnsof pixels P in images. The relative sizes of hands are indicated by the area Ai expressed as pixels. The hand's orientation and location are recognised by the middle of the hand characteristic. Determining the object's centre is crucial for identifying any variations in the form, location, or mobility of hands/fingers. Centres of hands are paired values (r_i, c_i) of rows and columns andmeasured as follows:

$$r_i = \frac{1}{A_i} \sum_{r=0}^{n-1} \sum_{c=1}^{n-1} rM_i(r, c) \quad (9)$$

$$c_i = \frac{1}{A_i} \sum_{r=0}^{n-1} \sum_{c=1}^{n-1} cM_i(r, c) \quad (10)$$

Where r and c imply rows and columnsof pixels P in images while $M_i(r, c)$ implies size functions. Distances between two open/closed fingers are used in computation of finger locations. The gesture is better defined by the finger's distance characteristic. One method of doing so is to count each consecutive pixel with 0 value up to a neighbouring picture element of value 1. One way to compute the significant points is as follows:

$$S_p = (avg\ x, avg\ y) \quad (11)$$

$$S_p = \left(\frac{1}{5}\sum_{p=sx}^{p=x+5}, \sum_{p=sy}^{p=y+5} + b\right) \quad (12)$$

where S_p represents significant points plotted on planes averaging 5-pixels on x and y axes, s_x implies starting points along x axes averaging subsequent 5 values, s_y represents starting points on y axes averaging subsequent values of 5 pixels.

Furthermore, by initialising non-zero random values, bias b adds another parameter for neural networks to adjust. Following combinations of collected features to create feature vector sets F_s, the data is displayed as symmetric patterns. To gauge hand's sizes, directions, and boundaries for certain gestures, which are described as

$$F_s = [Perimeter, Hand\ Size, Center\ of\ Hand, Finger\ Position \quad (13)$$

Where

$$Perimeter\ F_1 = [a_1, a_2, a_3, \dots, a_n] \quad (14)$$

$$Hand\ Size\ F_2 = [b_1, b_2, b_3, \dots, b_n] \quad (15)$$

$$Center\ of\ Hand\ F_3 = [x_1, x_2, x_3, \dots, x_n] \quad (16)$$

$$FingerPosition\ F_4 = [y_1, y_2, y_3, \dots, y_n] \quad (17)$$

Here "n" is the total number of extracted "feature points" for the hand gesture's defined features (F1, F2, F3, and F4).Hands' circumferences, centres, finger locations, and sizes can all be obtained by symmetric feature sets where size features extract features for comparing scale variations of hand, centres of hands handle hand orientations, finger positions measure distances between two fingers, and perimeters outline hands' physical shapes. These generated feature sets in matrix forms display all characteristics and integrated in Equation (18).

$$X = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \Rightarrow \begin{bmatrix} a_1\ a_2\ a_3 \dots a_n \\ b_1\ b_2\ b_3 \dots b_n \\ x_1\ x_2\ x_3 \dots x_n \\ y_1\ y_2\ y_3 \dots y_n \end{bmatrix} \Rightarrow \begin{bmatrix} 0\,0\,0\,1\,1\,0 \dots 0 \\ 0\,0\,1\,1\,0\,0 \dots 1 \\ 0\,0\,1\,1\,1\,0 \dots 0 \\ 0\,1\,1\,1\,0\,0 \dots 1 \end{bmatrix} \quad (18)$$

where the values of each matrix member are represented by pixels whose values are represented by "0" and "1," where "0" denotes lack of feature points and "1" indicates existence of feature points. Lastly, DSURF method is applied. In a variety of lighting scenarios, DSURF extracts features while identifying important focal points of SL based images along with their pertinent attributes.

Feature extractions are completed following this phase. The extracted feature vectors, or descriptors, and the locations that correspond to them are returned in this stage. The size of this feature vector, nx64, is identified through the number of retrieved feature points, n. The IGNN classifier will now be used to classify this feature vector.

C. IGNNbased Classification

Graph Neural Networks (GNNs) are becoming quite prominent in many sectors these days.The significant advancement in the field of graph analysis study is made possible by GNN's strength in simulating the relationships between nodes in a network. One particular type of neural network that works directly with the graph format is the GNNs [26]. Node categorization is a typical use case for GNN. As every node in the network has a label, estimating them without requiring any ground truth is the basic aim.

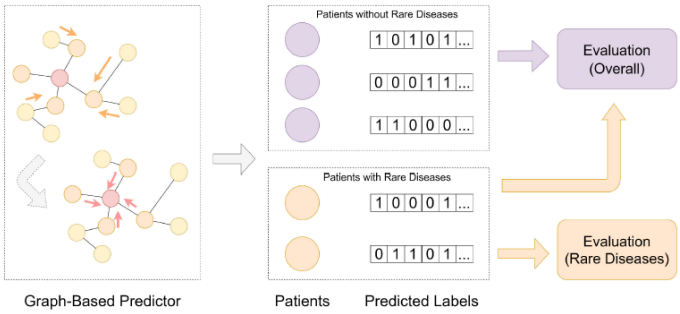


Figure 6: Illness predictions based on GNNs.

Figure 6 shows the method by which the suggested model classified hand motions by using particular attributes. Subsequently, the likelihood of hand gestures is gradually assessed by measuring the proximity of hand gesture embedding. The first paragraph of this section describes a graph encoder model z∈Rd for arbitrary graph nodes and generate low-dimensional embeds. An aggregation approach is used to identify each node's embedding by examining the attributes of its neighbouring nodes that are directly related to it, drawing inspiration from the recent breakthroughs in graph convolutional networks.The aggregation function supplied takes into account a node's first-order neighbours, and updates are applied to all nodes in the network. Nodes offer unique input fields for these computations; however, parameters defining distributions and interchanges of information are shared in all computations. This approach makes optimal use of the graph's dispersed information across regions and allows for the creation of embeddings during training for nodes that were not previously known, such recently joined individuals in the database.

Graph *G = {C, P}*, show hand gestures consistently *v∈ G* that are compact. Subsequently, *l*-th layers embed $h_v^l$of node *v* values and computed as:

$$h_{N(v)}^l = AGGREGATE\left(\left\{h_{v'}^{l-1}, \forall v' \in N(v)\right\}\right) \quad (19)$$

$$h_v^l = \sigma\left(W^l.\left[h_v^{l-1};\ h_{N(v)}^l\right]\right) \quad (20)$$

whereW^lrefer tothe weight matrix, whose learning is to be done at the *l*-th layer,h_v^{l-1}stands for nodes' *v*'s embeds at prior layers, andthe total layer sizes are represented by *L*. [.;.] implies concatenations of two vectors, and N(v) stands for collections of v's uniformly sampled neighbouring nodes. when l = 0, arbitrary counts or side data are utilized to initialise node embeds h0vRd. For example, h0v gets transformed into real-valued dense feature vectors, by digits reflecting detected values of feature dimensions using pre-defined hand motions like ages. The symbols represent synergic representations that arise from aggregations of embedding of nodes' v's neighbours at (l-1)-th layers $h_{N(v)}^l$. Using non-linear activations (such as

tanh) and aggregators like means, max. poolsi, RNNs, and so forth. The model's default,means ()aggregate data. Later, before obtaining final embeds of nodes at final layers L, normalisations are required:

$$h_v = \frac{h_v^L}{\|h_v^L\|_2}, \forall_v \in \mathcal{g} \text{ (21)}$$

It is crucial to highlight that in this work, the representations learnt using the general concept graph C and the embedding space of the input data graph P are the same.Embedding of symptoms in both graphs are equivalent for symptoms p, which operate as useful fillers in a variety of networks linking people. Projecting each node embeds into same spaces prior to non-linear activations aligns their contexts better. In this procedure, learning about diseases, patients, and symptoms are done in three separate embedding spaces and parallelly as different types of nodes:

$$z_v = \sigma(Wh_v), \forall_v \in \boldsymbol{g} (22)$$

Where z_v indicates the node v's ultimate embedding, while W stands for the learnable projection weights. It cannot, however, handle edge information (different edges in a knowledge network, for example, may imply different relations among nodes). Furthermore, a static point would not be optimal for learning during node presentation since it does not allow for a wide range of node propagation.Thus, in order to address the aforementioned problem, a Patrice Swarm optimisation technique is proposed in this technological paper.

Particle Swarm Optimization (PSO)

One of the particles utilised by PSO is a swarm that roams the search zone in order to discover the best answer [25].Particles may be seen as points in D-dimensional spaces, and their "flying" governed by their flight histories along with remaining particles's flight histories. The particles in D-dimensional spaces must travel at specified speeds to provide optimal results.

The description of the particle's I velocity is$V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$,the particle's position is expressed by$(x, x_{i2}, \ldots, x_{iD})$,the best place for particle I to be was given as$p_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$,another name for it is$p_{best}$.

All particles are at their maximum possible position globally when$p_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$,another name for it is$g_{best}$. Particles in groups have fitness functions to determine fitness values ab. Velocity updates for dimensions d are provided by Equations(23) and (24) in traditional PSO:

$$v_{id} = w \times v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times Rand() \times (p_{gd} - x_{id}) \text{ (23)}$$

$$(X_{id} = x_{id} + v_{id}) (24)$$

The following are the PSO parameters: "w" denotes "inertia weight," "Q" stands for population quantity, "C1 and C2" for acceleration constants, "v_{max}" for maximum velocity, "G_{max}" for "greatest amount of iterations," and "rand" and "Rand" for "random functions" with values in the range of [0,1". The values of C1 and C2 are typically calculated using constant 2.

EPSO improves data transmission through populations and preserves population variety during the optimisation procedure to get around the drawbacks of conventional optimisation algorithms and handle multiparameter, tight coupling, and nonlinear engineering optimisation challenges. Enhanced convergences and tendencies to reach local optimisations are a few of the drawbacks. "Local-global information shares" are chosen as performance parameter selections once parameters utilised in imported data have been assessed. When tested subsequently using many sets of conventional functions to confirm the EPSO's global search performance, both the EPSO and conventional optimisation strategies yielded good results.

The aim of this study is to introduce an ePSOform that aims to enhance the PSO algorithm's performance in obtaining better solutions while preserving its simplicity and quick convergence. In order to increase the algorithm's strength in both the discovery of new search spaces holding optimum solutions and the exploitation of intermediate solutions, this collision factor depends on the presentation of a common yet effective innovative operation throughout the iterative search process. The proposed version begins with the modified PSO variant that depends on parameter values.

Collision factor

Since the sizes corresponding to a text feature vector in text classification are typically quite large, the particles in PSO will collect to the point where finding the global optimum is still impossible..In order to guarantee optimal convergence, PSO is supplemented by the collision factor K. Equation (25) provides the expression for velocity:

$$v_{id} = K[v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times Rand() \times (p_{gd} - x_{id})] (25)$$

The equation above is used in this study to calculate the collisionfactor K. When Clerc's experiment was used, the identical value of 2.05 was used to obtain values c1 and c2.Here, K is the experimental variable with four decimal places. Equation (26) provides the precise velocity expression as follows:

$$v_{id} = 0.7298 \times [v_{id} + 2.05 \times rand() \times (p_{id} - x_{id}) + 2.05 \times Rand() \times (p_{gd} - x_{id}) (26)$$

During the first iterations, a particle in PSO can do recognition through a sizable time period in order to determine the likely location of the perfect solution. It must extend locally for a small amount of time in order to identify the optimum place for the subsequent repeats.K will thus need to use a higher number at the beginning and a lower number at the end. In the later phase, K must similarly gradually drop over an extended period of time, reaching its minimum. The concave function is demonstrated to be congruent with this fluctuation pattern.

Collision factor's initial iterations must select convex functions that allow particles to arrive at ideal solutions over extended periods of time for avoiding early convergences. Concave functions must be used for advanced iterations for allowingcollision factors to progressively decrease to lowest values for local developments. This ensures that the algorithm will eventually converge. In accordance with this idea, Equation (27) provides the functional collision factor that is created using the cosine function:

$$K = \frac{\cos((\pi/G_{max})\times T)+2.5}{4} \quad (27)$$

T standds for iteration counts. Initialize G_{max}= 40, to accommodate K's changing curves. Initially having a convex function, K curves eventually becomes concave. Values of K are substituted in Equation (27), and later Equation (27) varies in Equation (28) expressed below:

$$v_{id} = \left(\frac{\cos((\pi\times T/G_{max}))\times 2.5}{4}\right) \times [v_{id} + 2 \times rand() \times (p_{id} - x_{id}) + 2 \times Rand() \times (p_{gd} - x_{id})] \quad (28)$$

EXPERIMENTAL RESULTS

A functional testbed wascreated to validate the schema. Microsoft Kinect sensor captured images and turned them into depth images ofvarious resolution for data collection.

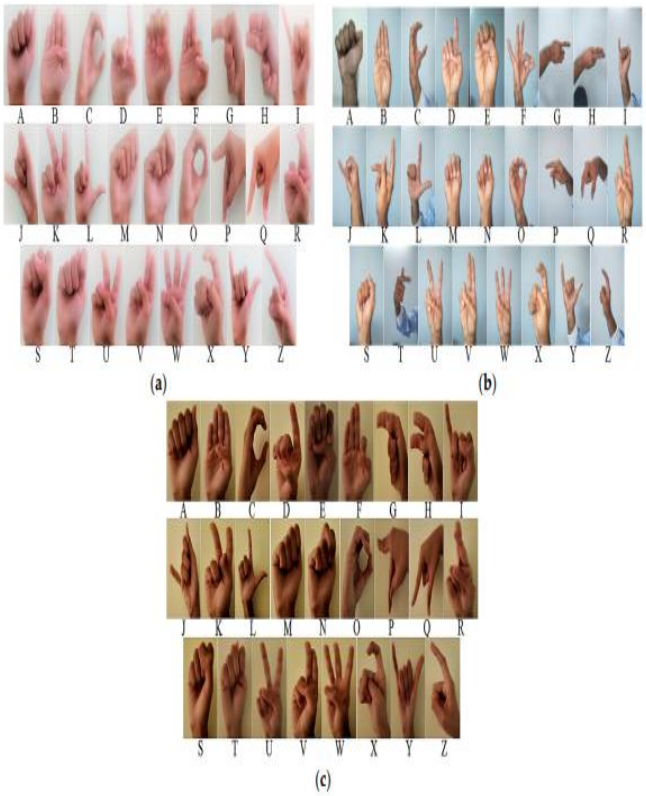


Figure 7: Data acquisition of SL Images under different lighting scenarios (a) bright (b) ambient (c) dark

Successive stage is to turn SL images into grayscale images for segmenting hand items from their backdrops, as illustrated in Figure 8. The next step is to use Equation (15) to determine the feature points by computing the DSURF points from the segmented hand objects. The significant image points from the selected feature points are processed using a feature descriptor approach to create significant vector points. To do this, the necessary feature point values are extracted from the 26 alphabetic letters using the Matlab programme, and they are then transformed into significant points by applying the DSURF method.

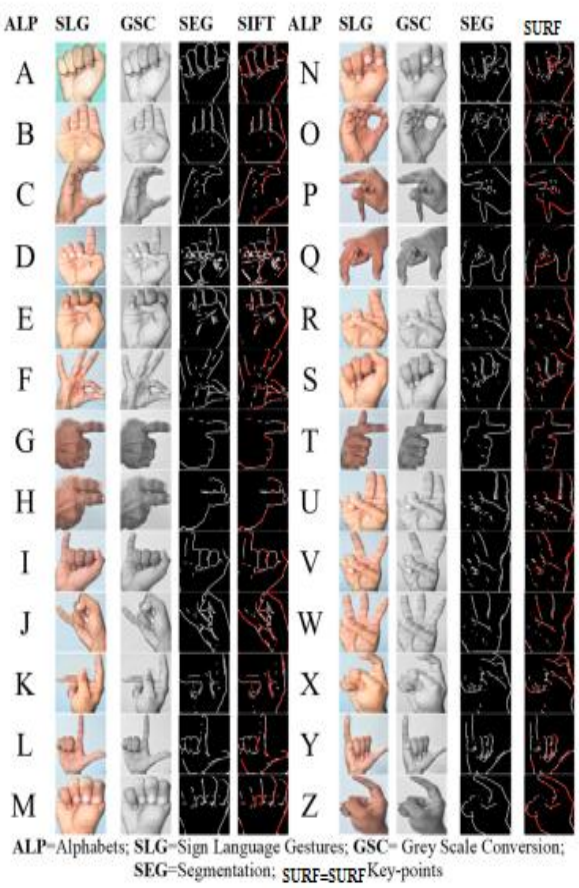


Figure 8: SL conversion into grey scale and segmentation

Following the SIFT calculation of SP values, the average processing time for each step is determined for various illumination settings and resolution proportions, as indicated in Table 1. Table 1 shows that, under ambient light circumstances, tolerable processing times are achieved at resolutions of 1024 × 768, which art good resolutions for analyses. It has been observed that processing times increase with increasing resolution rates. As a result, in the subsequent step,1024 x 768 resolutions are considered

Table 1: Average processing time under variant lightening condition at different resolution

Resolution	Average Processing Time (sec)		
	Bright Light	Ambient Light	Dark Light
260X175	2.08	2.00	2.02
320X240	2.11	2.01	2.32
640X480	2.41	2.12	2.16
800X600	2.02	2.04	2.34
1024X768	2.12	2.25	2.25
2048X1540	2.11	2.29	2.28
4160X3120	2.22	2.41	2.75

Four letters are taken into consideration to demonstrate the effectiveness of the suggested framework after the 26 alphabets are converted into grayscale, segmented, and important point calculated for A–Z. As seen in the accompanying Figure 9, the letters "P," "A," "I," and "R" are employed for segmentation, DSURF, and measuring the important locations of the hand motion.

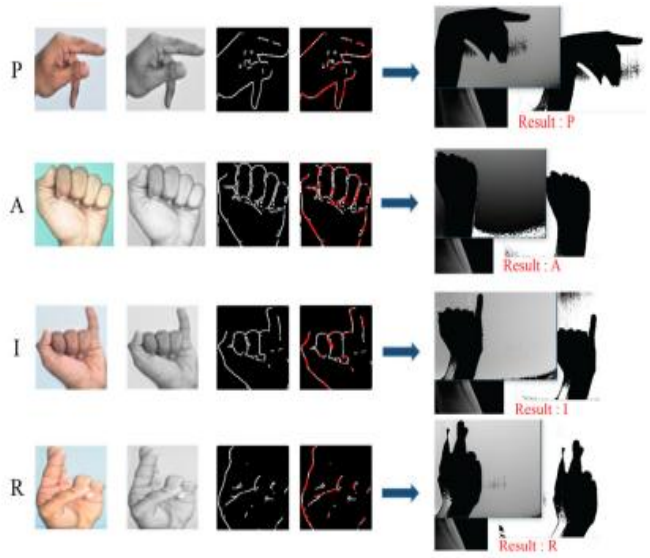


Figure 9: Case study feature and significant point's calculation using DSURF

Table 2 presents comparison of many parameters related to lighting conditions, including frame rates, pixel densities, gesture counts, classification algorithms, overall accuracies, and failure rates. Using preprocesses, PFE, picture segmentations, extractions of important keypoints, and selections of EGNN designs for classifications, high recognition rates in gesture recognitions were attained.

Table 2: Preformation comparison of the proposed and existing approaches

Approaches	# of Gesture	Test image	Frame Resolution	Recognition Time (s)	Accuracy (%)	Error Rate (%)|
CNN	8	195	320X240	0.11	93.9	6.1
DC-CNN	10	400	128X128	0.4	94.8	5.2
ANN	26	800	1024X768	0.013	97.4	2.6
EGNN	26	800	1024X768	0.010	98.1	2.2

CONCLUSION

This research introduced an effective framework that uses the luminosity technique to enhance gesture recognition performance under varying light. For usage in gesture identification, symmetric patterns and their corresponding luminosities of filters are considered. The four primary stages of the suggested framework are feature extraction, classification, image preprocessing, and picture acquisition. In order to create the datasets, a functional testbed is established in the lab employing two Microsoft Kinect sensors to record depth photographs in order to gather a variety of resolution data. SL-based gesture pictures are transformed into PNG format and kept in the database. The subsequent stage involves pre-processing the obtained photos to convert the collected image into a consistent brightness level. In order to do it, the system converts the input image to greyscale and applies the luminosity approach. Image segmentation difficulties are among the many activities that benefit greatly from grayscale conversion, which also makes it much easier to deal with. The weighted approach was used to convert to greyscale. Extracting the relevant traits and choosing them is the next step. Because more relevant characteristics need more space and computing time, this phase is crucial. To identify and extract the relevant characteristics from gathered data, the DSURF approach is provided. From a given input picture, the suggested approach extracts four important properties (finger distance, hand size, centre of hand, and perimeter). Subsequently, the characteristics that were extracted are integrated to create a feature vector set that is used to determine the hand's size, direction, and border for a specific motion. Next, under various lighting circumstances, the DSURF algorithm is used to pinpoint the important focal points of SL-based photos together with their pertinent properties. From recognised feature points, the significant image points are calculated and converted into significant vector points using a feature descriptor approach. Based on the findings, we can see that 1024 × 768 resolution is an acceptable resolution rate for analysis and has a fair processing time.

REFERENCES

[1] Zhigang, F., 1999, "Computer gesture input and its application in human computer interaction," Mini Micro Syst., 6, pp.418–421.

[2] Mitra, S., and Acharya, T., 2007, "Gesture recognition: A survey," IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., 37, pp. 311–324.

[3] Ahuja, M.K., and Singh, A., 2015, "Static vision-based Hand Gesture recognition using principal component analysis," Proceedings of the 3rd International Conference on MOOCs, Innovation and Technology in Education (MITE). Amritsar, India. 1–2 October 2015. pp. 402–406.

[4] Kramer, R.K., Majidi, C., Sahai, R., and Wood, R.J., 2011, "Soft curvature sensors for joint angle proprioception," In 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA. 25–30 September 2011. pp. 1919–1926.

[5] Jesperson, E., and Neuman, M.R., 1988, "A thin film strain gauge angular displacement sensor for measuring finger joint angles," Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society; New Orleans, LA, USA. 4–7 November 1988; pp. 807–vol.

[6] Fujiwara, E., dos Santos, M.F.M., and Suzuki, C.K., 2014, "Flexible optical fiber bending transducer for application in glove-based sensors," IEEE Sens. J., vol. 14, pp. 3631–3636.

[7] Shrote, S.B., Deshpande, M., Deshmukh, P., and Mathapati, S., 2014, "Assistive Translator for Deaf & Dumb People," Int. J. Electron. Commun. Comput. Eng., vol. 5, pp. 86–89.

[8] Gupta, H.P., Chudgar, H.S., Mukherjee, S., Dutta, T., and Sharma, K., 2016, "A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors," IEEE Sens. J., vol. 16, no. 16, pp. 6425–6432.

[9] Murthy, G.R.S., and Jadon, R.S., 2009, "A review of

vision based hand gestures recognition," Int. J. Inf. Technol. Knowl. Manag., vol. 2, no. 2, pp. 405–410.

[10] Khan, R.Z., and Ibraheem, N.A., 2012, "Hand gesture recognition: A literature review," Int. J. Artif. Intell. Appl., vol.3, no. 4.

[11] Suriya, R., and Vijayachamundeeswari, V., 2014, "A survey on hand gesture recognition for simple mouse control," Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, India. 27–28 February 2014. pp. 1–5.

[12] Sonkusare, J.S., Chopade N.B., Sor R., and Tade S.L., 2015, "A review on hand gesture recognition system," Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India. 26–27 February 2015. pp. 790–794.

[13] Kaur H., and Rani J., 2016, "A review: Study of various techniques of Hand gesture recognition," Proceedings of the 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India. 4–6 July 2016. pp. 1–5.

[14] Wang, Z., Chen, B., and Wu, J., 2018, "Effective inertial hand gesture recognition using particle filtering based trajectory matching," J. Electr. Comput. Eng.., pp. 1-9.

[15] Suguna, R., and Neethu, P.S., 2017, "Hand gesture recognition using shape features," Int. J. of Pure and Applied Mathematics., vol. 117, no. 8, pp. 51-54.

[16] Marium, A., Rao, D., Crasta, D.R., Acharya, K., D'Souza, R., 2017, "Hand gesture recognition using webcam. Am. J. Intell. Syst., vol. 7, pp. 90–94.

[17] Al Delail, B., Bhaskar, H., Zemerly, M.J., and Werghi, N., 2018, "Balancing Incident and Ambient Light for Illumination Compensation in Video Applications," In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018., pp. 1762–1766.

[18] Yu, W., Lei, B., Ng, M.K., Cheung, A.C., Shen, Y., and Wang, S., 2021, "Tensorizing GAN with High-Order Pooling for Alzheimer's Disease Assessment," IEEE Trans. Neural Netw. Learn. Syst., vol. 33, no. 9, pp. 4945–4959.

[19] You, S., Lei, B., Wang, S., Chui, C.K., Cheung, A.C., Liu, Y., Gan, M., Wu, G., and Shen, Y., 2022, "Fine Perceptive GANs for Brain MR Image Super-Resolution in Wavelet Domain," IEEE Trans. Neural Netw. Learn. Syst., pp. 1–13.

[20] Gozzi, N., Malandri, L., Mercorio, F., and Pedrocchi, A., 2022, "XAI for myo-controlled prosthesis: Explaining EMG data for hand gesture classification," Knowl. Based Syst., vol. 240.

[21] Wang, Y., Ren, A., Zhou, M., Wang, W., and Yang, X., 2020, "A novel detection and recognition method for continuous hand gesture using fmcw radar," IEEE Access., vol. 8, pp. 167264–167275.

[22] Xi, L., Chen, W., Zhao, C., Wu, X., and Wang, J., 2020, "Image Enhancement for Remote Image plethysmography in a Low-Light Environment," In Proceedings of the 15th IEEE International Conference on Automatic Face and Gesture Recognition, Buenos Aires, Argentina, 16–20 November 2020., pp. 761–764

[23] Pin, X., Chunrong, Z., Gang, H., and Yu, Z., 2020, "Object Intelligent Detection and Implementation Based on Neural Network and Deep Learning," In Proceedings of the 2020 International Conference on Computer Engineering and Application ICCEA 2020, Guangzhou, China, 18–20 March 2020., pp. 333–338.

[24] Bay, H., Tuytelaars, T., and Van Gool, L., 2006, "SURF: Speeded Up Robust Features," In Computer Vision—ECCV 2006, Leonardis, A., Bischof, H., Pinz, A., Eds.; Springer: Berlin/Heidelberg, Germany. pp. 404–417.

[25] Rublee, E., Rabaud, V., Konolige, K., and Orb, G.B., 2017, "An Efficient Alternative to SIFT or SURF., In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011., pp. 2564–2571.

[26] Tareen, S.A.K., and Saleem, Z., 2018, "A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," In Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 3–4 March 2018., pp. 1–10.