

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications



ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

A novel static and dynamic hand gesture recognition using self organizing map with deep convolutional neural network

K. Harini & S. Uma Maheswari

To cite this article: K. Harini & S. Uma Maheswari (2023) A novel static and dynamic hand gesture recognition using self organizing map with deep convolutional neural network, *Automatika*, 64:4, 1128-1140, DOI: [10.1080/00051144.2023.2251229](https://doi.org/10.1080/00051144.2023.2251229)

To link to this article: <https://doi.org/10.1080/00051144.2023.2251229>



© 2023 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 29 Aug 2023.



Submit your article to this journal [↗](#)



Article views: 721



View related articles [↗](#)



View Crossmark data [↗](#)



A novel static and dynamic hand gesture recognition using self organizing map with deep convolutional neural network

K. Harini and S. Uma Maheswari

Department of ECE, Coimbatore Institute of Technology, Coimbatore, India

ABSTRACT

Gesture recognition has gained a lot of popularity as it allows humans to communicate with real or virtual systems through gestures, offering new and natural interaction modalities. Recent technologies, such as augmented reality (AR) and the Internet of Things (IoT), have witnessed enormous growth in computer applications that focus on human–computer interaction (HCI). However, a few of these tactics make use of a combination of methods, such as image segmentation, pre-processing, and classification. The hessian-based multiscale filtering and YCbCr colour space are used to separate the gesture region to be recognized. A modified marker-controlled watershed method is employed to segment the gesture contour along with the eight-connector graph to increase recognition precision. The proposed hand gesture recognition methodology uses Self Organizing Map (SOM) with Deep Convolutional Neural Network (DCNN) provides better results with fast convergence speed. Experiments were carried out on a dataset of 30 static and 6 dynamic gestures and also evaluated on a publicly available IITTA-ROBITA ISL Gesture Database to show the effectiveness. The results show that the suggested method can recognize gesture classes with 95.63% accuracy rate without significantly affecting the recognition time. The proposed algorithm was then implemented to control household appliances.

ARTICLE HISTORY

Received 19 April 2023
Accepted 18 August 2023

KEYWORDS

hand gesture recognition; human–computer interaction; hessian-based multiscale filtering; modified marker-controlled watershed algorithm; eight-connected filling algorithm; SOM-based deep convolutional neural network

1. Introduction

Researchers in the field of body language have identified a wide variety of nonverbal cues, including posture, hand gestures, and facial expressions, that convey meaning. Therefore, human physical language, and more specifically the use of hand gestures, are critical factors in direct human-to-human interaction. Every communication process involves the use of hand gestures, and understanding these can shed light on the nature of language. Static gestures use a single image for processing at the input of the classifier whereas in contrast, dynamic gestures use image sequences for processing. Basically, there are five kinds of gestures. Deictic or indexical gestures are also known as pointing where children extend their index finger. Motor gestures or beat gestures usually consist of short, repetitive, rhythmic movements that are closely tied with prosody in verbal speech. Lexical or Iconic gestures are full of content, and may echo, or elaborate, the meaning of the co-occurring speech. Metaphoric gestures put an abstract idea into a more literal, concrete form. The most familiar are the so-called emblems or quotable gestures. These are conventional, culture-specific gestures that can be used as replacement for words. A single emblematic gesture is capable of conveying meaning independent of speech. Some good examples are the thumbs up, the peace sign, and the middle finger. The

use of hand gestures in place of vocalizations characterizes Indian sign language alphabets and digits in the live stream, as in ref [1]. Sign language is a system of communication in which each sign represents a different word or letter of the alphabet with a predicted output label in the form of text. Gesture variations in the face and hand, head, and body movements are the building blocks of sign language. Human–Computer Interface, video games, virtual reality, domestic appliances, automation, body language, etc. are just some of the many areas where a gesture recognition system can be put to use. Implementation of gesture recognition and design issues for deaf, hard of hearing, and dumb people were discussed. However, it is costly and difficult to locate qualified interpreters for their regular interactions throughout their lives [2].

There are many different sign languages used in different parts of the world. Sign language is based on the local culture and language. People who are deaf or mute in India use Indian Sign Language (ISL) to communicate [3]. The gestures can be broken down into two main categories: static and dynamic. Postures, strokes, pre-strokes, stages, and reactions are all included in dynamic gestures, providing a thorough review of the state-of-the-art techniques used in hand gestures. It is made up of both stationary and moving movements. Dynamical gestures include the movement of body

components, while static gestures merely involve the adoption of a particular position. According to a survey conducted by the Indian government in 2011, more than 26.8 million Indians are living with a disability. 19% of these people have some kind of speech impairment, and 7% have some kind of hearing impairment. The SLR system is useful for bridging the gap between hearing individuals and the deaf community. Body language in ISL is more complex, making it harder to accurately track and segment hands. ISL is a common language that bridges the gap between the hearing and the hearing-impaired.

Using machine learning techniques is necessary for this purpose. To accomplish this, the dataset can be used to build models or algorithms, after which they can be classified or labelled to make a forecast. Several cutting-edge algorithms, including the histogram of oriented gradients (HOG), the convolutional neural network (CNN), and bagging, could produce excellent outcomes. Some classification and output-generating techniques like Naive Bayes (NB), Support Vector Classifier (SVC), Logistic Regression, K-Nearest Neighbour (KNN), and Stochastic Gradient Descent (SGD) can also be employed [4]. The use of vision-based recognition does not demand any specialized hardware and instead relies on a web camera or a depth camera. Specialized tools such as wired or wireless wraps monitor the actions of the user with hand and gesture sensing consoles such as Microsoft Kinect, Leap Motion, etc., that preserve the hand gestures and motions. Figure 1 represents the basic steps of hand gesture recognition.

Although the classical methods show promise, they are unable to construct reliable descriptors for hand position detection in real-time applications [5]. These problems stem from the inability of traditional machine learning methods to correctly discern patterns in unaltered input data. The detection and segmentation of hands from photos acquired under complex background settings [6] is one such issue encountered by the hand posture recognition approach. Another challenging aspect is identifying the strong elements that describe the mathematical variations in the presentation of a similar hand posture by different people. CNNs are frequently employed to solve issues involving spatial data, such as photographs. When processing temporal,

sequential data, like text or films, RNN (Recurrent Neural Network) perform better.

In the proposed work, the gesture area is segmented using a YCbCr colour space skin colour filter, and the noise around the segmented skin colour area is processed using hessian-based multiscale filtering. The gesture area is then segmented using a modified watershed algorithm based on a marker, and the gesture area is filled using an eight-connected filling algorithm. A new deep architecture based on stacked recurrent neural networks has been developed. To effectively describe local patches of the input image using SOM neuron coordinates, this part takes advantage of the topological order property of the SOM feature map.

The paper is organized as follows: Section 1 describes the introduction to the research work, the problem statement, and the contributions of the work with respect to the current state-of-the-art in the field. Section 2 discusses the most recent works that are related to the proposed gesture recognition method. Section 3 explains in detail the proposed robust image representation method using the SOM-DCNN network. Section 4 shows the proposed SOM-DCNN model along with results and performance metrics. Section 5 provides the experimental setup. Section 6 of this paper offers the conclusion of the proposed work along with suggestions for further research.

2. Related works

The research reviewed several techniques for acquiring hand gestures using a variety of acquisition tools, some of which relied on images captured instantaneously and others on previously recorded images from databases. The acquisition process can be classified as either static, in which the gestures are indicated by a consistent image, or dynamic, in which the gestures are depicted by the motion of the hand, which is an additional way of categorizing the data. Zhao et al. [7] presented a dynamic time-warping DTW algorithm and recurrent neural network running on an interactive wristband with computational resource requirements as low as Flash. Temporal Multi-Modal Fusion (TMMF) was introduced by Gammulle et al. [8] as a single-stage

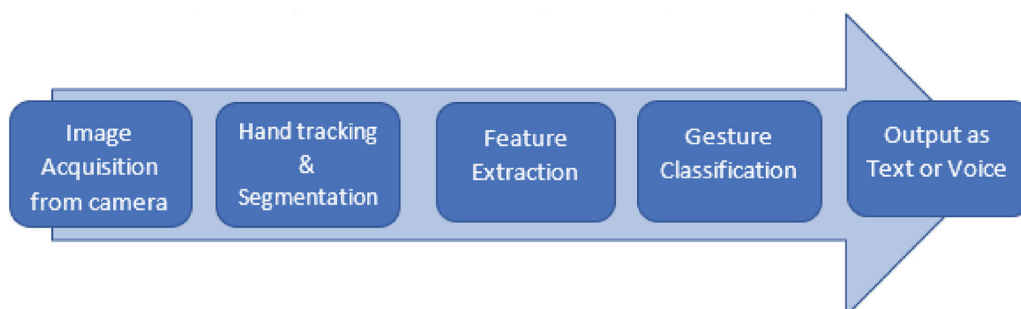


Figure 1. Basic flow of hand gesture recognition.

continuous gesture recognition framework that can recognize and classify numerous gestures in a video using a single model. This method learns the natural transitions between gestures and non-gestures with models for mapping unimodal feature mapping (UFM) and multimodal feature mapping (MFM).

Utilizing multiple deep learning architectures for hand segmentation, local and global feature characterizations, and sequence feature modernization and recognition, Al-Hammadi et al. [9] proposed a new system for dynamic hand gesture recognition. This demonstrates that the system is superior to current methods. Zhang, X., and Wu, X. [10] used dynamic and static gesture recognition to come up with an innovative algorithm to control a robot. Kolla Bhanu Prakash et al. [11] proposed using a deep convolutional neural network to automatically label images and modified recurrent neural network (RNN) models for hand gesture assessment to identify hand motion and object interaction. Hsieh and Liou [12] found the user's face and centred the image on the face and upper torso. The ability of dynamic up, down, left, and right-hand gestures to be classified using motion history images and four novel Haar groups is investigated. Like features in the presence of a complex background, it provides an average accuracy of 95.37% with 3.93 ms processing time per frame.

The method proposed by Patil and Subbaraman [13] and described in this study seeks to recognize dynamic hand gestures by employing Hough transform-based spatiotemporal feature extraction to preprocess and an artificial neural network for identification, with an average Recognition Rate (RR) of over 94% on the Cambridge database and over 98% on the Sebastien database. Hand feature extraction is described by Liu et al. [14], and the corresponding hand gesture recognition method is presented. A Linear Discriminant Analysis (LDA) algorithm was issued to process the feature vectors that are calculated by aligning a series of Concentric Circular Scan Lines (CCSL) with the palm's equator. Neethu et al. [15] used a CNN classification approach to identify human hand gestures. Enhancement techniques like adaptive histogram equalization are used to increase the contrast of each pixel in an image. Nayak et al. [16] suggested an enhanced memetic firefly algorithm is used to optimize the Light Gradient Boosting Machine (GBM) hyper-parameters with a 99.36% accuracy rate and a high degree of stability. A CNN model trained with deep learning was developed for sign language recognition by Sharma and Singh [17]. VGG-11 and VGG-16 have also been developed and tested in this work to help evaluate the model's performance.

Tan et al. propose a specialized network architecture. The Enhanced Densely Connected Convolution

Neural Network (EDenseNet) disseminates the features being used again to all the extracted features in a bottleneck manner, as well as having the Conv layer that comes after it smooth out the unwanted features, thereby further strengthening feature propagation, which achieves an average accuracy of 99.64%, both of which are better than other deep learning-driven instances. Using an effective deep CNN architecture, Adithya and Rajesh [18] proposed a method for the identification of hand gestures, the primary component in sign linguistic knowledge. Improved recognition accuracy was demonstrated using two publicly available datasets, including ASL datasets and one NUS hand gesture dataset. Convolutional neural networks have made great strides in the field of deep learning in recent years [20,21]. In ref. [22], a method was proposed to classify word symbols using the neuro-fuzzy approach and natural language processing (NLP) technology to display the final word. In [27] S. Kanade et al., principal component analysis (PCA) features and support vector machines (SVM) were used to construct a system using a bespoke dataset that had good accuracy. SVM and HOG features are used into build a sign language recognition system using data from single-handed signs. Also, [24] reviews single-handed or simple hand movements. In this example, we used single- and double-handed custom datasets with two different data collection methods in this example. [27,28] Uses A deep convolutional neural network provides enhanced hand gesture recognition techniques for both static and dynamic gestures compared to various existing methods. Sun et al. [29] proposed an end-to-end Three-Branch Embedding Network (TBE-Net) for the identification of similarities in vehicles. In ref. [30], the real-time small object detection (RSOD) algorithm improves the accuracy of small object detection. Table 1 shows the research gap of related work along with its benefits and drawbacks.

Most current gesture recognition technologies are built for either static or moving motions. Typically, the Kinect sensor is chosen to capture movements in order to collect more depth information, and it has proven to have a notable performance of light sensitivity in gesture identification. However, there are still problems with gesture detection, and segmenting using depth data requires further attention. This research aims to perfect a real-time, low-complexity, and efficient method for identifying static and dynamic gestures performed with a single or many hands from depth data. The primary objective is to use cutting-edge deep learning techniques to correctly classify hand gestures according to their proper definitions with the highest degree of accuracy possible. A novel strategy for the same is being developed, and it will be compared to several distinct broad standard models.

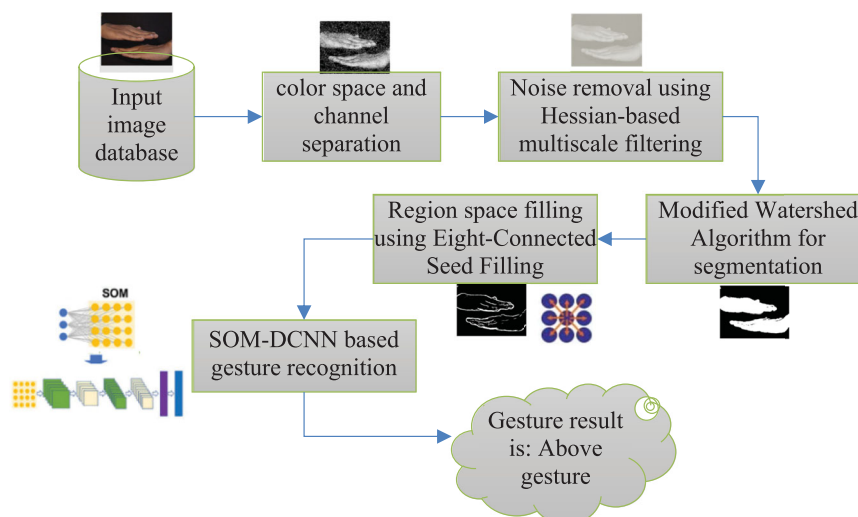
Table 1. The research gap of related work with their advantages and disadvantages.

Author	Method	Advantages	Disadvantages
Gammulle et al., [8]	TMMF	As a result, the language model may become overly precise.	These, however, are labour-intensive to execute and not always reproducible.
Al-Hammadi et al., [9]	multiple deep learning architectures	The suggested system makes efficient use of both local and global configurations, giving special focus to the hand area.	However, this type of system is only effective when dealing with simplified alphabets and numerals, which only depend very slightly on the global setup, and not when dealing with genuine sign language motions.
Bao et al., [11]	deep convolutional neural network	Hand localization in such settings may be difficult due to the lack of readily available ground truth bounding boxes for use in training.	The system may make a mistake or perform poorly if it cannot determine where two signs begin and end.
Hsieh & Liou [12]	Support Vector Machine	The method works well and produces high accuracy	Most visual hand motion detection systems are limited to "controlled situations", where lights and complicated backgrounds are absent.
Patil & Subbaraman [13]	Neural Network	In contrast to most other video processing systems, this one doesn't need to worry about detecting the foreground.	Edges in the system will scatter the growing amount of noise.
Liu et al., [14]	W-KNN	The effectiveness and precision of gesture recognition were greatly enhanced by using this technique.	This approach is quite sensitive to background noise.
Neethu et al., [15]; Sharma & Singh [17]; Adithya & Rajesh [18]	CNN	By connecting appearances of hands and fingers across frames, tracking improves the features' temporal trajectories.	If the object or body being extracted is too huge, mistakes will be made.
Nayak et al., [16]	LightGBM	Keeping tabs on things allows you to keep the model-based estimations up-to-date.	Nevertheless, there is a slight compromise to be made when deciding between computational expense and precision.
Tan et al.,	EDenseNet	This approach uses supervised learning to modify itself for new circumstances.	Achieving adequate generalization on unseen data, however, is dependent not just on the EDenseNet's architecture, but also on the volume and diversity of the training data.
Malek et al., [19]	K-NN based Algorithm	Using K-NN based Algorithm for Video Annotation Purposes	Analyses a classroom video as an input, and then extracts the vocabulary of twenty gestures. Uses K-NN algorithm to determine the hand gesture

3. Proposed methodology

The purpose of the method is to quickly and accurately recognize gestures while also reducing the likelihood of capturing accidental motions and minimizing the effects of interference. Figure 2 depicts the expected results of the suggested algorithm. The model does image processing instantly. The primary goals before the gesture data are to reduce noise in the data images,

establish strong gesture features, standardize the data scale, streamline the training process, and enhance identification precision. Pre-processing steps used in this study include a skin colour detector, a modified marker-based watershed algorithm [22,24], an eight-connected seed-filling approach, and a scale normalization technique. Using multiple convolutional SOM layers, the proposed SOM based Deep CNN (SOM-DCNN) network is a novel deep learning architecture.

**Figure 2.** Proposed framework for hand gesture recognition.

The common CNN shares some structural similarities with this. When it comes to dynamic and static hand motion recognition, however, each convolutional SOM layer is trained independently after the training of the preceding layers has concluded.

3.1. Input database

Human-robot interaction researchers have recently become interested in the challenge of learning to decipher Indian Sign Language gestures. To spur future development and study in the field of gesture recognition, IIITA – ROBITA will distribute the ISL Gesture Database to researchers for free. Since July 2009, the information has been gathered in the Robotics and AI Lab, IIIT-Allahabad. 23 unique motions were recorded at 30 frames per second on a Sony Handy camera with a constant background and varying levels of lighting [20,21]. A total of 60 gesture images were used for the training and validation of the proposed model. Sequences of RGB frames for 24 ISL gesture databases,

along with 26 Indian Sign Language (ISL) alphabets and 10 gesture words for both static and dynamic gestures, were used as database images. Table 2 represents the gesture images in the IIITA-ROBITA ISL gesture database for both static and dynamic gestures. Table 3 represents the ISL alphabet and a few word gestures in the proposed system database.

3.2. Database image pre-processing

The gesture image is improved using Hessian-based multiscale filtering, which involves mixing a Hessian matrix with a Gaussian convolution to adapt the filtering response to the individual scales. Images from the IIITA-ROBITA ISL dataset are used in the new model. Both complete and partial datasets are evaluated. The analysis of the Eigen values of the scale space of the Hessian matrix is the foundation for the filter [22]. The intensity and orientation of blood flow can be inferred from the eigen values and eigenvectors of the Hessian matrix HMX . The Hessian matrix $H(x, y)$ for a gesture

Table 2. Samples of gesture images of IIITA-ROBITA ISL gesture database Static and Dynamic.

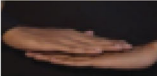

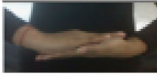
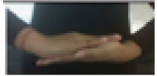





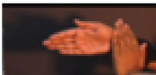
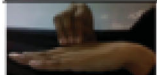



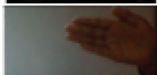

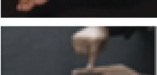
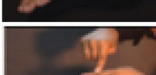


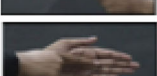

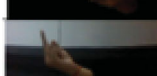
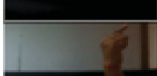
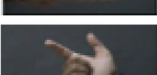
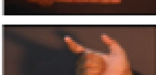
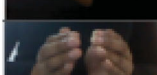
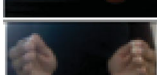
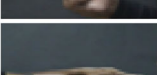
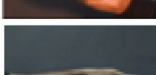
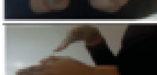

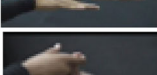
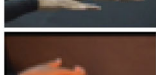
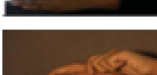

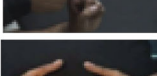
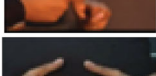
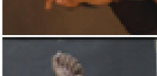
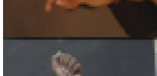

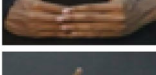


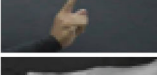
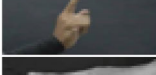
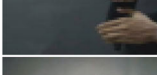
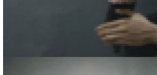


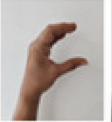





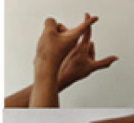


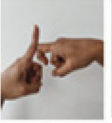
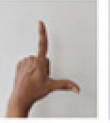






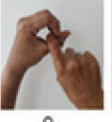
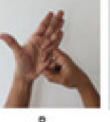
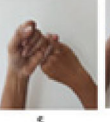
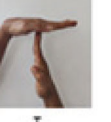

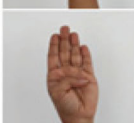



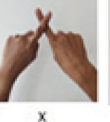
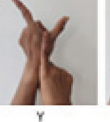







Static gesture sign	Start frame	End frame	Dynamic Gesture sign	Start frame	End frame
Above			Add		
All Gone			Bring		
Flag			Color		
Across			Depart		
Middle			Distant		
Beside			Come		
Moon			Big		
Advance			Dive		
Keep			Afraid		
All Gone			Ascend		
Alone			Bag		
Below			Yes		

Table 3. ISL alphabets and few word gestures in database.

Statics Gestures						Dynamic Gestures		
						H		
A	B	C	D	E	F		J	
						Walk		
G	I	K	L	M	N		Follow	
						Thank you		
O	P	Q	R	S	T		Help	
								
U	V	W	X	Y	Z			
								
Gift	Call	Stop	Smile	Wrong	Time			

$GI(x, y)$ is a 2×2 matrix made up of second-order partial derivatives of the gesture image GI , as in Equation (1); The formula for $HMX(x, y)$ is:

$$HMX(x, y) = \begin{pmatrix} \frac{\partial^2 GI}{\partial x^2} & \frac{\partial^2 GI}{\partial x \partial y} \\ \frac{\partial^2 GI}{\partial x \partial y} & \frac{\partial^2 GI}{\partial y^2} \end{pmatrix} \quad (1)$$

Like the gesture image GI , the Hessian is discrete but can be approximated to a continuous function by use of the 2-dimensional Gaussian filter (2DGF) as given in Equation (2) and the convolution differentiation property in Equation (3).

$$2DGF(x, y, \sigma) = \frac{1}{\sqrt{(2\pi\sigma^2)^3}} \exp^{-\frac{1}{2} \left(\frac{x^2}{\sigma_1^2} + \frac{y^2}{\sigma_2^2} \right)} \quad (2)$$

$$\begin{aligned} HM(x, y) &\approx 2DGF(x, y, \sigma) * \begin{pmatrix} \frac{\partial^2 GI}{\partial x^2} & \frac{\partial^2 GI}{\partial x \partial y} \\ \frac{\partial^2 GI}{\partial x \partial y} & \frac{\partial^2 GI}{\partial y^2} \end{pmatrix} \\ &= \begin{pmatrix} \frac{\partial^2 GF}{\partial x^2} & \frac{\partial^2 GF}{\partial x \partial y} \\ \frac{\partial^2 GF}{\partial x \partial y} & \frac{\partial^2 GF}{\partial y^2} \end{pmatrix} * GI(x, y) \quad (3) \end{aligned}$$

Where the convolution kernel is a Gaussian with scale σ , denoted by $GF(x, y, \sigma)$. Consider $HM(x, y)$ to have

eigen values $|\omega_1| \leq |\omega_2|$, ϵ_1 and ω_2 , Ev_1 and Ev_2 are real numbers that are the corresponding Eigenvectors, and $*$ is the convolution symbol. Since $|\omega_1|$ is the smallest eigen value, it corresponds to the smallest curvature eigenvector, Ev_1 , and since $|\omega_1| \cong 0$ is the greatest eigen value, it corresponds to the largest curvature eigenvector, Ev_2 . This means that the radial axis of the blood artery is parallel to Ev_2 , whereas the longitudinal axis of the blood vessel is parallel to 1. Using these values, two different measurements were devised to evaluate the anisotropy and contrast of the pixel. Equations (4) & (5) provide the formulas for calculating these values. Anisotropy is the first ratio. For a blob-like structure, \mathbb{R}_{anist} can account for the deviation, but it cannot tell the difference between a line-like and a plate-like pattern. To differentiate between plate-like and line-like features, the second contrast ratio, \mathbb{R}_{cont} , is used.

$$\mathbb{R}_{anist} = \frac{|\omega_1|}{|\omega_2|} \quad (4)$$

$$\mathbb{R}_{cont} = \sqrt{|\omega_1|^2 + |\omega_2|^2} \quad (5)$$

The likelihood that a pixel is part of a tumour throughout the classification process increases as \mathbb{R}_{anist} decreases for any given pixel. If there isn't enough contrast, \mathbb{R}_{cont} will be modest, hence a high number indicates a higher likelihood that the pixel in question is part of a non-gesture. Curvature C will be negative, or $|\omega_2| < 0$, for photos in which the regions are darker

than the background, resulting in malignancies that look like valleys. Based on these findings, a likelihood function as in Equation (6) was created for each scale \mathfrak{S} ,

$$C_0(\mathfrak{S}) = \begin{cases} 0 & \text{if } |\omega_2| > 0 \\ e^{-\frac{\mathbb{R}_{anist}^2}{2t1^2} \left(1 - e^{-\frac{\mathbb{R}_{cont}^2}{2t2^2}} \right)} & \text{otherwise} \end{cases} \quad (6)$$

where $t1$ and $t2$ are thresholds that determine how sensitive the line filter is to the measurements \mathbb{R}_{anist} and \mathbb{R}_{cont} , respectively.

3.3. Segmentation using modified marker-controlled watershed algorithm

To fix the problems with the standard watershed algorithm, a modified version called the marker-controlled watershed algorithm is proposed, and its block diagram is depicted in Figure 3. This model studied a morphological technique to clean up photographs and then selects the foreground and background markers, which handle the issue of over-segmentation and enhance the watershed segmentation yield with a higher precision rate [23].

This technique also involves flooding the imaged landscape to produce catchment basins and watersheds. Instead of directly using the gradient image for flooding, adaptive thresholding is implemented on it. To finish, at this part superimpose the markers on the gradient threshold image. The main difference is that not all minima are employed to initiate absorption in the gradient field, but rather just a few endpoints. Now, the foreground marker, background marker, and gesture boundaries are identified using several morphological techniques, such as attempting to open, trying to

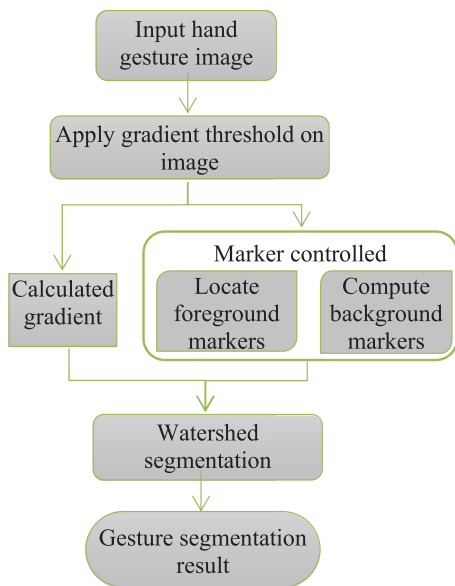


Figure 3. The process diagram of the modified marker-controlled watershed algorithm.

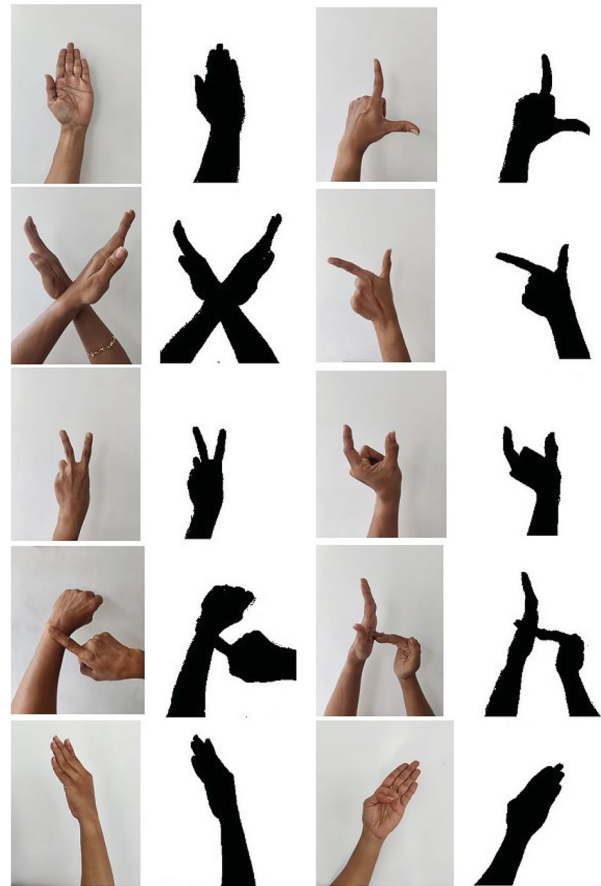


Figure 4. Sample gestures before and after segmentation.

close, degradation, dilation, restoration, and thresholding methods. The final watershed segmented output of the original image may be created using this method, where the gesture area of the hand is extracted from the original hand image.

Figure 4 shows some sample images of gestures before and after segmentation process. To identify the gesture area on an image, the hand areas are partitioned using global thresholding. None of these frames are helpful; in fact, the majority of them are non-informative frames with pre- or post-gestures. The critical frames for each gesture must be extracted, for this reason. The key motions are determined using the gradient value of each frame. Given that hand gestures and key gestures differ significantly in size and intensity when compared to the black backdrop, the adaptive thresholding segmentation could locate key gestures with sufficient accuracy. A series of many frames combine to produce a single gesture. The key motions are determined using the gradient value of each frame. One typical application of watershed segmentation is to recover homogenous or very homogenous intensity regions or blob-like objects. In practice, watershed segmentation is typically used on gradient photographs rather than the original images because a region with little intensity changes has a tiny gradient [31].

The goal of this stage is to provide images of gradient magnitude for use in the segmentation phase that follows. To do this, the filtered image GI from the pre-processing step is subjected to the application of two Sobel filters, one horizontal and the other vertical as in Equation (7). The two edge pictures produced by the procedure are G_x and G_y . Gradient image GRI is then calculated using these images.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +3 \end{bmatrix} * GI$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * GI$$

$$GRI = \sqrt{G_x^2 + G_y^2} \quad (7)$$

The foreground and background marker positions are initially mapped using the gradient magnitude image. The gradient magnitude image should then be altered so that the only locations with regional minima are the foreground and background markers. The segmentation of watersheds is then applied to the updated gradient magnitude image. The eight-connected seed-filling algorithm [24] is an improvement over the four-connected filling algorithm. Whereas the four-connected filling algorithm starts from a point of injection in the neighbourhood and extends in four directions, encompassing all of the area's pixels, the eight-connected seed-filling algorithm speeds up the process by extending in eight directions. The scale normalization operation may guarantee the uniformity of extracting features and thereafter label the gesture data as predictable and stable for neural network training. The segmented and filled picture data are normalized in this article, which can significantly accelerate convergence, increase model accuracy during training, and reduce gradient explosion.

4. Proposed SOM-DCNN model for gesture recognition

For the deep learning network to accurately predict hand gestures, it is fed a dataset of hand gestures in this work. Convolutional layers are used to extract the best features from images, and SOM layers are used to organize the most important features of the images.

By adjusting the most crucial hyper parameters like the SOM layer's dimension, patch size, block size, the number of neurons, and initial neighbourhood radius, you can analyze how the proposed SOM-DCNN feature extraction method performs under different conditions. As can be seen in Figure 5, SOM-DCNN is composed of the following layers: input layer, SOM layer, two convolution layers, two relu layers, two maxpooling layers, a fully connected layer, and an output layer.

4.1. The SOM layer

In most cases, the input variables for the K neurons that make up the SOM can be given the form of random vectors. Next, apply the learned convolutional M layer to the input image to calculate a representation \mathfrak{N} for each $w \times w$ patch. Therefore, map all patches using the neighbourhood radius function to establish the feature index image as the output depiction of the input image. When given an image GI with $n \times n$ pixels, it is possible to derive a set of $(n - w + 1) \times (n - w + 1)$ feature benchmark images with N channels. The N coordinate values of the winning neuron W are used to construct the representation \mathfrak{N}_{ij} for the ij patch of the input image. An essential part of the SOM learning algorithm is determining which unit is the best fit for each input sample. In addition to its position on the map, each neuron in the SOM layer is also associated with a mean vector or mv . Minimizing the Euclidean distance between the centre of each neuron in the SOM

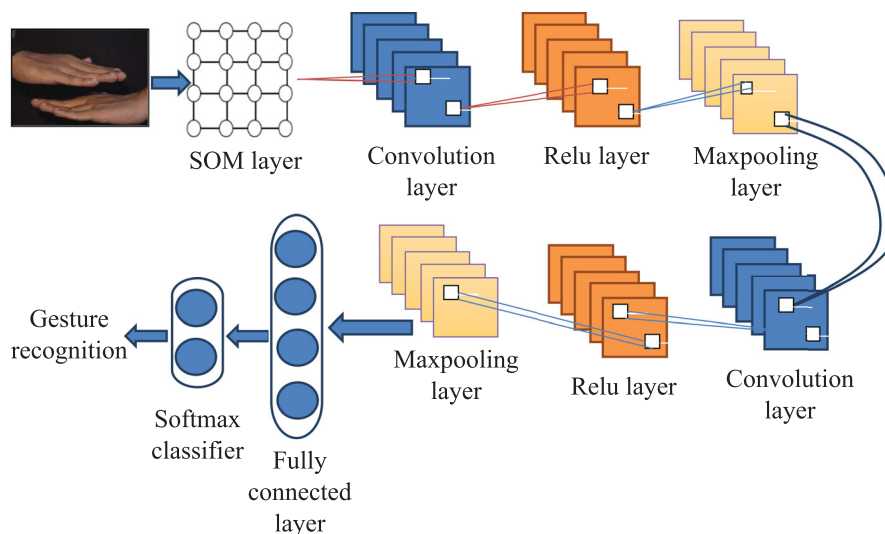


Figure 5. Proposed framework diagram of SOM-DCNN-based gesture recognition.

grid and the input image patch $x(t)$ pinpoints the \mathfrak{W} .

$$\mathfrak{W} = \arg \min_i x(t) - mv_i(t)^2 \quad (8)$$

At each iteration, the entire X training data set (70% of data input image database) is used to recalculate the map's parameters. Each sample's \mathfrak{W} is calculated using Equation (8). Here is an updated version of the mean vectors for all neurons ($i = 1, \dots, K$) as in Equation (9). The expression for the velocity of an object at time $t + 1$ is:

$$mv_i(t + 1) = \sum_t \frac{NF_{\mathfrak{W}i} * x(t)}{NF_{\mathfrak{W}i}} \quad (9)$$

where $NF_{\mathfrak{W}i}$ represents the NF of the i th neuron given \mathfrak{W} . In the manner shown by Equation (10), the \mathfrak{W} affects all of its spatial neighbours in the map grid. Selecting a different neighbourhood function may result in different amounts of change for each neuron. When it comes to self-organization, the neighbourhood function is crucial. A popular option for the NF is a Radial Basis Function (RBF) with a radius of ζ .

$$NF_{\mathfrak{W}i} = \mathcal{L}(t) e^{\frac{(-\mathfrak{W} - i)^2}{2\zeta^2(t)}} \quad (10)$$

where \mathfrak{W} and i are the coordinates of the \mathfrak{W} – winner neuron and the i -neuron, respectively, in the SOM map grid, and $\mathcal{L}(t)$ is a monotonically linear decreasing function of t representing the learning rate. Their dimensions are related to the size of the SOM map. For each iteration t , the neighbourhood value is determined by plugging in the neighbourhood radius function $\zeta(t)$, into Equation (11). The formula for $\zeta(t)$, is:

$$\zeta(t) = \zeta_i + \frac{t}{T}(\zeta_f - \zeta_i) \quad (11)$$

where ζ_i , ζ_f represent the beginning and ending radii, respectively. The function $\zeta(t)$, decreases linearly with the iteration count T . Training begins with a large value for the initial radius ζ_i and decreases to a small value ζ_f by the end. At the outset of training, the topological order of the neurons is established, and by the end, it has converged to nearly optimal values. It is common practice to use local patches extracted from N feature index images as input for the subsequent training session of the second convolutional layer. The SOM-DCNN method is used to recognize the gesture regions of various static and dynamic images in the DCNN model [26]. The layered DCNN is used to analyze the input images for a variety of gestures. The DCNN examines the layers, which are the input images, to derive the regions to classify. Convolution layers are responsible for the maximum pooling of the input image subsamples, and this is where these samples are convolved. As a result, the output layer is defined by a set of networks that are all fully connected.

4.2. Convolutional layers (CONV)

To describe it simply, a deep convolutional neural network (DCNN) is comprised of many layers of convolutional neural network (CONV). Each of the K programmable filters (“kernels”) that make up the CONV layer's parameters has a width and a height and is nearly always a square. The size of these filters is negligible, but they permeate the entire volume. The number of image channels used as input to a CNN is referred to as “depth” (i.e. a depth of three when working with RGB images, one for each channel). Thus, each value in the output volume represents the “look” of a single neuron at a discrete part of the input. By doing so, the network “learns” to activate filters whenever a certain class of feature appears in a particular region of the input volume. When filters in the network's lower layers detect edges or corners, they may activate. High-level features, such as facial features, a dog's paw, a car's hood, etc., may then trigger filters in the network's later layers. The idea of activation is like neurons getting “excited” and “activating” when they detect a known pattern in an input image. This research, at this part, used the ReLU activation function for all activations because of its robust expression ability and freedom from the vanishing gradient problem, which allowed us to keep the model's convergence rate stable. Here $f(x)$ is the ReLU formula as in Equation (12). When x is an integer,

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \quad (12)$$

To prevent further over fitting and parameters of lower networks, the Max pooling layer can eliminate extraneous features by down sampling them. The “pooling layer” can be expressed using the following Equation (13);

$$x_j^l = f(\omega_j^{l*} \text{down}(x_j^{l-1}) + b_j^l) \quad (13)$$

where ω_j^l represents the weight of the j th feature map in the l th layer and b_j^l represents the bias in that map. In statistics, down-sampling (*) stands for the mean, the maximum, and the stochastic pooling functions. Max-pooling with shift-invariances was used to reduce the output feature map dimension.

4.3. Fully connected layer (FC)

After several different convolution and pooling processes, the final FC layer can learn either the output class or the input sample probability. The mathematical expression for the FC-layer as in Equation (14), in which all k th-layer neuron nodes are connected to all $k-1$ th layer output nodes:

$$y^k = f(wc^k x^{k-1} + b^k) \quad (14)$$

where k is the layer number in the network, y^k was the output of the FC layer, x^{k-1} was the unfolded eigenvector in dimension one, wc^k was the weighted coefficient, b^k was the bias, and f^* , the activation function of the final FC layer, was a Softmax function for classifications.

4.4. Performance evaluation

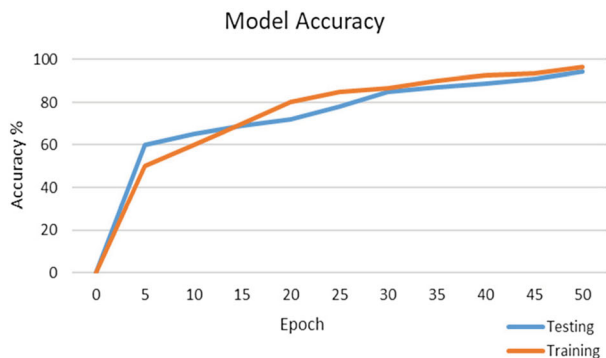
Precision, sensitivity/recall, specificity, f-measure, Peak Signal Noise Ratio (PSNR), Mean Square Error (MSE), and accuracy were used as evaluation measures to determine the efficiency of the proposed method in classifying ISL images into their respective categories. In this context, True Positives (TP) refer to correctly identified positive gestures, while False Negatives (FN) refer to incorrectly identified negative gestures. When gestures are perceived as negative, they are called True Negative (TN), while when they are perceived as positive, they are called False Positives (FP).

When compared to earlier approaches to classification, the contour has a higher rate of convergence. The proposed method efficiency was compared with three state-of-the-art classification methods for gesture recognition: support vector machines [23,25], deep convolutional neural networks [26], and EDenseNet [28]. During the training period, a total of fifty epochs were run to evaluate the accuracy and loss of the model, as shown in Figure 6(a) & (b). In addition, the testing dataset that was employed for both of these techniques was loaded with images of the gesture itself as well as some random images from the database. This was done to ensure that the results were accurate.

4.5. Sensitivity

Sensitivity measures the % age of actual positives values which are correctly classified and calculated as in Equation (15)

$$\text{Sensitivity/recall} = \frac{TP}{TP + FN} \quad (15)$$



(a)

4.6. Specificity

The % age of accurately identified negative values is a measure of specificity. The ratio of those who tested negative (TN) to those who are genuinely negative (TN + FP) is known as specificity (also known as True Negative Rate). When the patient does not have a brain disease, sensitivity can be thought of as the likelihood that the test would provide a negative result Equation (16)

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (16)$$

4.7. Precision

Proportion of correctly classified positive samples to the total count of positive predictions on samples and calculated as in Equation (17)

$$\text{Precision} = \frac{TP}{FP + TP} \quad (17)$$

4.8. F-measure

It is the harmonic mean of precision and recall, also called F_1 -score and calculated as in Equation (18).

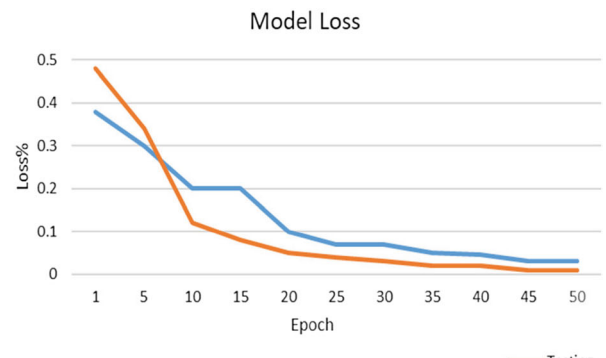
$$\text{F-measure} = \frac{2 * (\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})} \quad (18)$$

4.9. PSNR

The PSNR is used to find the deviation of classified image and from the ground truth image as represented in Equation (19)

$$\text{PSNR} = 10 \log_{10} \frac{R^2}{MSE} \quad (19)$$

where R is maximum fluctuation in input image data. For double precision data types, the value of R is 1, and for 8-bit unsigned data types, R is 255.



(b)

Figure 6. (a) & (b) Accuracy and loss graph for training and testing of SOM-DCNN model.

4.10. MSE

It is the error metric used to compare image classification quality. The lower the value of MSE, the lower the error and better the quality of classification. MSE is calculated as in Equation (20)

$$MSE = \sum_{M,N} \frac{[I_1(M,N) - I_2(M,N)]^2}{M*N} \quad (20)$$

4.11. Classification accuracy

Ratio of correctly classified samples to the total samples count as in Equation (21)

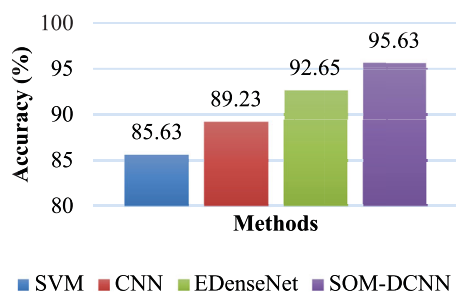
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (21)$$

4.12. Performance analysis

For the performance evaluation, the datasets are divided into 70% to train and 30% to test. Various performance metrics to evaluate the proposed model has been done and is compared with other existing approaches. Table 4 represents the comparison table of the proposed SOM-DCNN model with other existing approaches like SVM, CNN, and EDenseNet. It is observed that the proposed model provides better accuracy with less computational time when compared with all other existing approaches with an accuracy of 95.63% with an average computational time of 5.2

Table 4. The numerical results of proposed and exiting methods with performance metrics.

Metrics	Model performance			
	SVM [24,27]	CNN [28]	EDenseNet [30]	SOM-DCNN
Precision (%)	85.15	87.22	89.57	91.26
Recall/ Sensitivity (%)	83.42	85.15	87.43	90.98
Specificity (%)	85.65	88.34	90.27	92.44
F-Measure (%)	84.27	86.17	88.49	91.19
PSNR (dB)	35.78	38.61	42.44	45.39
MSE	04.22	03.75	03.56	03.22
Accuracy (%)	85.63	87.11	89.65	92.63
Computation time (sec)	07.50	06.50	05.60	05.20



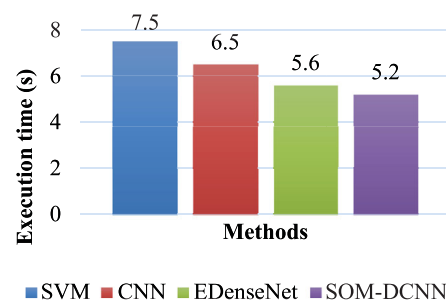
(a)

secs. The accuracy and computational time plots of the proposed model in comparison with other existing approaches is shown in Figure 7(a) & (b) respectively.

5. Experimental results and discussion

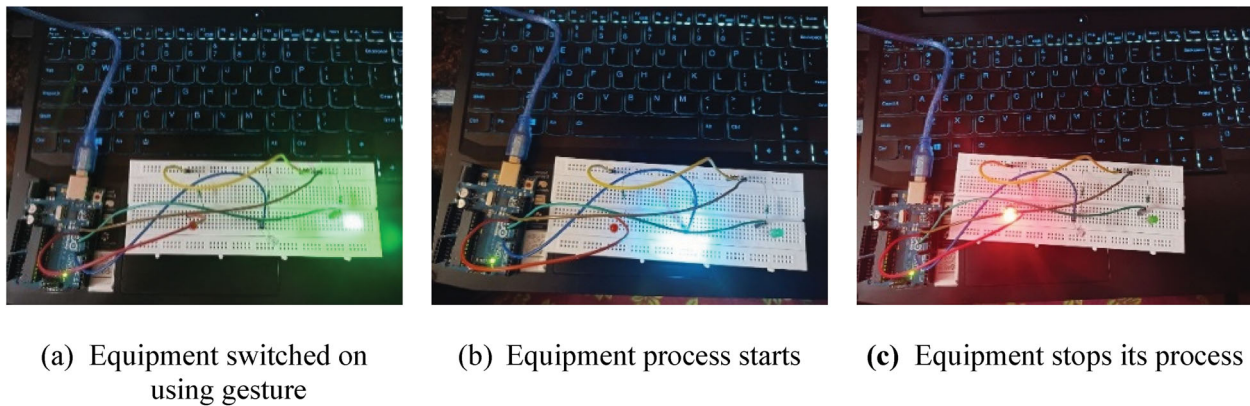
Experiments utilizing SOM-DCNN classification were performed using a hand gesture controller that makes it possible to control appliances using hand tilt angle values. The hand gesture controller actually uses an accelerometer to function, and it can be used anywhere a tilt is available for managing and directing operations. A wireless XBee transmitter connected to the laptop transmits data to a second XBee receiving platform. After receiving the data, that platform processes it before sending it to the main controller, which contains a PIC chip. To achieve the desired result, it communicates a variety of directives to the entire system. Using a PIC18 Microcontroller acts as the nerve centre of the project and as a tool with the ability to easily interface with the Wi-Fi module in this situation. A 32-bit microcontroller model is being used, which runs at a speed of 40 MHz. Both the self-balancing robot and the transceiver analyzer were developed and implemented as part of this work. The sensor measurements from the gyroscope are sent to the microcontroller, which uses the SOM-DCNN technique to process them. In this way, the device is instructed to move around the surface and discover what has to be changed in order to produce a balanced output for gesture detection.

By using hand gesture controller-based gesture detection, a data collection of 60 ISL gestures – 42 static gestures and 18 dynamic gestures is created in three different backdrop situations. The matching control signal that is transmitted from the gesture unit detects the motion of the person's fingers and controls the robot unit. Flex sensors and an XBee-S1 assist the PIC microcontroller within the gesture unit in reading gestures. The following are the implementation steps: To start the process, click the "Start" button (the one with the green light) and follow the on-screen directions. This task will load the gesture test in the subsequent stage and



(b)

Figure 7. (a) & (b) Classification accuracy comparison and computation time comparison results between existing and proposed method.



(a) Equipment switched on using gesture

(b) Equipment process starts

(c) Equipment stops its process

Figure 8. Hardware circuit board and process for gesture recognition.

then determine the gesture feature's value. The computed value is sent to the Arduino in the next phase, which acts as an interface between the software and the hardware. The signal is then sent to the relay, which acts as a switch, and its content is determined by the feature's value and by identifying the gestures using SOM-DCNN. The feature value of household appliances is used to regulate them. The last step is to stop the process (Red light). The hardware circuit board and process are illustrated in Figure 8(a), (b) & (c)

6. Conclusion and future work

This study uses skin colour detection, a modified marker-based watershed algorithm, and a seed-filling algorithm to pre-process the gesture data so that it better fits the deep learning of gesture recognition. The identification success rate of recognition test set data can reach 95.63% under natural light conditions through the training of 60 gesture data after pre-processing by un-supervised deep CNN trained with self-organizing maps (SOM-DCNN). To learn hierarchical features from training images, the network employs a cascade of convolutional SOM layers. The proposed approach is in line with the study of modelling increasingly complex and variable functions by combining the nonlinearities of neurons into networks. The SOM-DCNN network performs as well as other state-of-the-art methods on gesture datasets that have been polluted with random and image background noise.

The primary objective of gesture recognition research as it relates to Human-Computer Interaction (HCI) is to develop systems that can recognize certain human gestures and utilize them to communicate information or operate devices. Robotic arms and other equipment of a similar nature can be operated using hand gesture recognition technologies. The proposed method can be extended and implemented for various home appliances through Internet of Things (IoT) for gestures-controlled home automation/ office automation in future. The proposed algorithm can also be

implemented in various other applications like automatic vehicle identification & control and also in identification of autistic children using pose estimation. Although the proposed work provides better results for dynamic gestures, recognition of sequence of gestures to identify as a sentence/ statement were unable to implement. Hand tracking in cluttered background for diversified gesture was quite challenging. The future work of this research may be implemented for long video streams to identify sign language.

Acknowledgements

The authors would like to thank the college management, principal, head of department for having permitted to carry out the research work.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

- [1] Katocha S, Singh V, Tiwary US. Indian sign language recognition system using SURF with SVM and CNN. *J Array*. 2022;14:1-9.
- [2] Nagpal N, Mitra A, Agrawal P. Design issue and proposed implementation of communication Aid for deaf & dumb people. *Int J Rec Innov Trends Comp Commun*. May 2021;3(5):147-149.
- [3] Cheok MJ, Omar Z, Jaward MH. A review of hand gesture and sign language recognition techniques. *Int J Mach Learn Cybern*. August 2017;10:131-153. DOI:10.1007/s13042-017-0705-5
- [4] Bhushan S, Alshehri M, Keshta I, et al. An experimental analysis of various machine learning algorithms for hand gesture recognition. *Electronics*. 2022;11(6):1-15. DOI:10.3390/electronics11060968
- [5] Rautaray SS, Agrawal A. Vision based hand gesture recognition for human computer interaction: a survey. *Artif Intell Rev*. 2015;43(1):1-54. DOI:10.1007/s10462-012-9356-9
- [6] Zhang T, Lin H, Ju Z, et al. Hand gesture recognition in complex background based on convolutional pose machine and fuzzy Gaussian mixture models. *Int J Fuzzy Syst*. 2020;22:1330-1341. DOI:10.1007/s40815-020-00825-w

- [7] Zhao S, Cai H, Li W, et al. Hand gesture recognition on a resource-limited interactive wristband. *J Sci Technol Sens.* August 2021;5713:1–19.
- [8] Gammulle H, Denman S, Sridharan S, et al. Tmmf: temporal multi-modal fusion for single-stage continuous gesture recognition. *IEEE Trans Image Process.* 2021;30:7689–7701. DOI:10.1109/TIP.2021.3108349
- [9] Al-Hammadi M, Muhammad G, Abdul W, et al. Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. *IEEE Access.* 2020;8:192527–192542. DOI:10.1109/ACCESS.2020.3032140
- [10] Zhang X, Wu X. Robotic control of dynamic and static gesture recognition. In *Proceedings of the 2019 2nd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, Shanghai, People's Republic of China, 22–24 November 2019. p. 474–478.
- [11] Bao P, Maqueda AI, del-Blanco CR, et al. Tiny hand gesture recognition without localization via a deep convolutional network. *IEEE Trans Consum Electron.* 2017;63(3):251–257. DOI:10.1109/TCE.2017.014971
- [12] Hsieh CC, Liou DH. Novel Haar features for real-time hand gesture recognition using SVM. *J Real Time Image Process.* 2015;10(2):357–370. DOI:10.1007/s11554-012-0295-0
- [13] Patil AR, Subbaraman S. A spatiotemporal approach for vision-based hand gesture recognition using Hough transform and neural network. *Signal Image Video Process.* 2019;13(2):413–421. DOI:10.1007/s11760-018-1370-1
- [14] Liu Y, Wang X, Yan K. Hand gesture recognition based on concentric circular scan lines and weighted K-nearest neighbor algorithm. *Multimed Tools Appl.* 2018;77(1):209–223. DOI:10.1007/s11042-016-4265-6
- [15] Neethu PS, Suguna R, Sathish D. An efficient method for human hand gesture detection and recognition using deep learning convolutional neural networks. *Soft Comput.* 2020;24(20):15239–15248. DOI:10.1007/s00500-020-04860-5
- [16] Nayak J, Naik B, Dash PB, et al. Hyper-parameter tuned light gradient boosting machine using memetic firefly algorithm for hand gesture recognition. *Appl Soft Comput.* 2021;107:107478. DOI:10.1016/j.asoc.2021.107478
- [17] Sharma S, Singh S. Vision-based hand gesture recognition using deep learning for the interpretation of sign language. *Expert Syst Appl.* 2021;182:115657. DOI:10.1016/j.eswa.2021.115657
- [18] Adithya V, Rajesh R. A deep convolutional neural network approach for static hand gesture recognition. *Proc Comput Sci.* 2020;171:2353–2361. DOI:10.1016/j.procs.2020.04.255
- [19] Alksasbeh MZ, Omari A, Alqaralleh B. Smart hand gestures recognition using K-NN based algorithm for video annotation purposes. *Indones J Electr Eng Comp Sci.* 2021;21(1):242–252. DOI:10.11591/ijeecs.v21.i1.pp242-252
- [20] Nandy A, Mondal S, Prasad JS, et al. Recognizing & interpreting Indian sign language gesture for human robot interaction. In *Proc. ICCCT-10, IEEE Xplore Digital Library.* 2020, p. 712–717.
- [21] Shailesh B, Shubham D, Rohin C, et al. Sign language recognition using neural network. *Int Res J Eng Technol.* April 2020;7(4):583–586.
- [22] Hemina B, Jeegar T. Indian sign language recognition using framework of skin color detection, Viola-Jones algorithm, correlation-coefficient technique and distance-based neuro-fuzzy classification approach. *Emerging Technology Trends in Electronics, Communication and Networking*, July 2020, p. 235–243.
- [23] Huang H, Li X, Chen C. Individual tree crown detection and delineation from very-high-resolution UAV images based on bias field and marker-controlled watershed segmentation algorithms. *IEEE J Select Top Appl Earth Observ Remote Sens.* 2018;11(7):2253–2262. DOI:10.1109/JSTARS.2018.2830410
- [24] Ravi P. A review on image based Indian sign language recognition. *Int J Innov Res Comp Commun Eng.* December 2018;6(12):9101–9107.
- [25] Das A, Ghoshal D. Human skin region segmentation based on chrominance component using modified watershed algorithm. *Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)*, *Procedia Computer Science*, 89. p. 856–863.
- [26] Deshpande PD, Kanade SS. Recognition of Indian Sign Language using SVM classifier. *Int J Trend Sci Res Develop.* April 2018;2(3):1053–1058.
- [27] Molchanov P, Gupta S, Kim K, et al. Hand gesture recognition with 3D convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops.* 07–12 June 2015, p. 1–7.
- [28] De Smedt Q. Dynamic hand gesture recognition-From traditional handcrafted to recent deep learning approaches. *Sciences and Technologies, Université Lille 1*, 2017.
- [29] Prakash KB, Eluri RK, Naidu NB, et al. Accurate hand gesture recognition using CNN and RNN approaches. *Int J Adv Trends Comp Sci Eng.* 2020;9(3):3216–3222. DOI:10.30534/ijatcse/2020/114932020
- [30] Sun W, Dai L, Zhang XR, et al. RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring. *Appl Intell.* 2022;52(8):8448–8463. DOI:10.1007/s10489-021-02893-3
- [31] Yuan L, Yu Q, Shen C, et al. New watershed segmentation algorithm based on hybrid gradient and self-adaptive marker extraction. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. 2016, p. 624–628. DOI:10.1109/CompComm.2016.7924776.