International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Artificial Intelligence Based Sign Language Prediction by Using the Twin Delayed Deep Reinforcement Memory Network architecture

G.M. Karthick[1], P.Kirubanantham*[2], A. Saranya[3] , M.Sayeekumar[4]

Submitted: 12/11/2022 **Accepted:** 13/02/2023

Abstract: Communication between the hearing- and speech-impaired and the rest of society may be greatly aided by developments in Sign language recognition (SLR). One of the most important building blocks of sign language comprehension is word-level sign language recognition (WSLR). However, due to the fact that the meaning of a word relies on a wide range of subtle body gestures, hand configurations, and other behaviors, identifying signals in films may be difficult. Recent pose-based WSLR designs either represent the temporal information without completely utilizing the spatial information or explain the spatial but not the temporal correlations among the postures in various frames. To address the problem of WSLR, we use a novel approach based on AI to collect posture data and carry out recognition. To begin, we pulled the data and ran it through a Fibonacci ripple Chebyshev filter for preliminary cleaning (FRCF). Linear embedding Hessian component analysis (LEHCA) is then used to extract the features. The statistical looper wing butterfly optimization (SLWBO) approach is then used to segment the regions of interest. In the end, Twin Delayed Deep Reinforcement Memory Network (TDDRMN) architecture explicitly analyzes the feature interactions and recognizes the meaning of sign language to aid in the decision making process. The results derived on WLASL, a typical word-level sign language recognition dataset, show the system's superiority over the traditional approaches by obtaining high accurate prediction with minimum time in decision making.

Keywords: *Sign language recognition, Artificial intelligence, Fibonacci ripple Chebyshev filter, Linear embedding Hessian component analysis, statistical looper wing butterfly optimization algorithm, Twin Delayed Deep reinforcement memory network*

1. Introduction

Video recordings of sign language (SL) are often used to spread awareness of the language used by the deaf community or for educational purposes. Common opinion is that SL has the most rigorously organized grammar of any sign language. Due to its unique characteristics, SL recognition has garnered significant interest in the fields of multimedia, artificial intelligence (AI), and computer vision (CV) as a place to investigate solutions to issues like human motion analysis, HCI, and UI design.

Many similarities between pattern recognition and computer vision are shown in the topic of joint research known as "behavioral recognition," which includes the recognition of sign language. "Sign language recognition" describes the entire procedure of observing human gestures, identifying the representations, and translating them into instructions with semantic meaning.

Fig. 1. Different kinds of signs

Detection, tracking, and recognition are the three processes that are involved in the classic techniques to sign language and gesture recognition. These stages are necessary to finish the majority of hand conversations. In addition, the major objective of the gesture recognition system is to locate the user's hands and then to segment the visual region that corresponds to those hands. This segmentation is crucial because it transfers only task-relevant information to the next

[1] *Vellore Institute of Technology –Vellore - 632014*
ORCID ID: 0000-0002-8425-7828
[2] *SRM Institute of Science and Technology, Kattankulathur - 603203*
ORCID ID: 0000-0002-7539-9293
[3] *SRM Institute of Science and Technology, Kattankulathur - 603203*
ORCID ID: 0000-0002-4340-4430
[4] *Vivekanandha college of engineering for women, Tiruchengode, Namakkal-637205*
ORCID ID: 0000-0003-2190-3009
** Corresponding Author Email: kirubanp2@srmist.edu.in*
** Corresponding Author Email: kirubanantham04@gmail.com*

stage, which concentrates on tracking and identification without passing along information about the visual background. Therefore, typical approaches are time-consuming and need several rounds of pre-processing to be accomplished.Because of this, the authors of this study developed a twin delayed deep reinforcement memory network using deep learning for SL detection. We propose a model comprised of four distinct modules: preprocessing, feature extraction, segmentation, and classification. We find that the limited variety of datasets prevents comprehensive training from making full advantage of the high-complexity deep neural network. In order to find a solution to this issue, we have been investigating an optimization approach that may successfully assist our classification architecture using segmentation. By using linear embedding and Hessian component analysis, we are able to directly control the process of segmentation and extract features from the data. After that, the region of interest (ROI) is segmented using a statistical looper wing butterfly optimization technique. Our AI-based deep learning network will be able to keep learning and benefit from the more nuanced gesture alignments after we deploy the Twin Delayed Deep reinforcement memory network. The following is a condensed version of the most important contributions that may be drawn from our work:

1. To achieve state-of-the-art performance on continuous SL recognition without importing extra supervisory information, we first develop our architecture with a Twin Delayed Deep reinforcement memory network with more learning capacity; second, we design an iterative optimization process for reducing the workload of our deep neural network architecture over time.

2. Third, we include a Twin Delayed Deep reinforcement memory network into our architecture, which is an improvement over the present state-of-the-art.

The following is the outline for the rest of this project's sections. Past studies that have investigated SL recognition are discussed in Section 2. In Section 3, we discuss the problem's formalization, and in Section 4, we show off our deep learning network for SL detection and the optimization strategy we used to get the best results. The experimental findings of the suggested procedure are discussed in Se**ction** 5, and the work is brought to a close in Section 6.

2. Related Works

Computer vision experts have spent the greater part of three decades studying SLR because of the problem's significance. The two most common types of SLR methods are initially isolated SLR, in which a single sign is identified at a time, and continuous SLR, in which many phrases are recognized over the continuously streaming input. The challenge of sign language gesture identification has prompted a number of different strategies. Many different approaches have been explored in the past in an effort to classify Second Life hand gestures.

The author of [1] proposes an attention method for learning to read Chinese hand signals. The researchers investigate the issue of lonesomeness by use of electromyography (EMG) armbands and multimodal data from the anterior cingulate cortex (ACC), gyral ruff (GYR), and spinal oculomotor (SMR) channels. In order to extract features in both the temporal and data domains, they based their network on a convolutional neural network (CNN-RNN) architecture. Incorporating an attention mechanism such as SE-Block, which helps to provide proper weight to different channels of information, greatly improves our strategy. Understanding the effects of temporal and spatial occlusion on sign language understanding is the focus of [2]. Video transformer model (VTN) results on the WLASL dataset are comparable to an I3D baseline, demonstrating that manual cropping may contain enough information for accurate prediction. CNN networks have been the subject of much study for their potential to resolve problems with image recognition and classification. Recognizing sign language has been an area of research and development over the last several years. It was suggested that a convolutional neural network (CNN) based approach, using a Gaussian skin color model and background removal, may be used for motion identification in camera pictures. By specifically filtering away the non-skin color of the picture and using the Gaussian skin color model to regulate the influence of light on skin color, we were able to attain an accuracy of 93.80% on a select dataset [3]. It was suggested to use a two-stage convolutional neural network (CNN) architecture (HGR-Net) [4]; the first stage would choose the area of interest by carrying out semantic segmentation at the pixel level. Recognizability is enhanced by 1.6% using the OUHands dataset when the suggested architecture is applied to the first stage, which combines a fully convolutional residual network with spatial pyramid pooling. To find the motion in RGB-D videos, a deep convolutional network (MultiD-CNN) was developed using a multidimensional feature learning approach [5]. Due to the usage of 3D ResNet to train a model with spatiotemporal features and the long short-term memory (LSTM) for understanding temporal correlations, the technique outperformed the prior methods on a broad variety of datasets. The dynamic graph approach to spatiotemporal attention (DG-STA) used by Chen et al. to recognize hand gestures was novel. The node characteristics and edges of the hand skeleton are learned using a fully connected network and a self-attention approach, with the computational cost being minimized by use of a new spatiotemporal mask. Experimental results show that DG-STA is superior to competing approaches for recognizing hand gestures [6]. A deep learning-based strategy using two ResNet CNNs with w attention with a fully linked layer was suggested for detecting moving objects. As an alternative, it was suggested that digital videos be compressed into a single RGB picture and then sent into the algorithm for final categorization. Experimental results using publicly accessible datasets [7] demonstrate that the suggested strategy outperforms the state-of-the-art in terms of accuracy. As a model for gesture recognition, we propose using Convolutional Neural Networks. Combining dynamic depth images (DDI), dynamic depth normal images (DDNI), and dynamic depth motion normal images (DDMNI) with bidirectional rank pooling yields valuable spatial and temporal data. At the 2016 ChaLearn LAP competition evaluation, the suggested model showed a 16.34% increase in accuracy [8] using the IsoGD dataset. When it came to the gesture detection problem, two distinct deep learning approaches were applied. The motion features of RGB sequences were revealed by using a convolutional two-stream consensus voting network (2SCVN) to explicitly define both the short-term and long-term structures. The suggested strategies enhanced accuracy by 4.47 percentage points compared to 2016's models when applied to the ChaLearn IsoGD dataset [9]. Based on a recurrent 3D convolutional neural network (CNN) model, the system created by the authors of [10] can recognize gestures in real time. RGB, depth, optical flow, and stereo IR data were combined to improve identification accuracy. Positive accuracy on the ChaLearn dataset is improved by 1% with the suggested model compared to the mean of all other models. It was suggested that the depth map and optical flow be utilized as inputs to two-stream convolutional neural networks (CNNs) for hand motion detection and identification. Compared to prior models using the identical MSR Action3D dataset [11], the suggested model achieves an accuracy improvement of 18.91%. The model for understanding sign language was developed by Rastgoo et al. They used a limited Boltzmann machine to analyze the photos (RBM). To build the model, we used data from three photos that included both color and depth (the unaltered original, the cropped version, and the noisy version). Each image's hand is first identified using a convolutional neural network (CNN), and then one of three versions of the identified hand is sent to the RBM in the form of RGB and depth channels. The output of the RBM will be combined with other signals to determine the name of the sign. The proposed model surpasses the state-of-the-art techniques [12], as shown by its performance on four publicly available datasets. In 2020, based on the RBM concept, the author of [13] presented a deep cascaded model for recognizing sign language in cinema. We were able to extract temporal aspects from the data by giving the LSTM a mixture of hand features, extra spatial hand relation (ESHR) features, and hand posture (HP) information. For this reason, the SSD model was also applied to the challenge of hand detection. Their suggested model was 4.25 percentage points more accurate than competing methods on the IsoGD dataset. Bag of Visual Words (BOVW) is presented in [14] as a model for recognizing the letters and digits of the Indian sign language alphabet (A-Z) and numerals (0-9) in a live video stream, with the anticipated labels being spoken out or shown on the screen. We use both skin tone and background removal in the segmentation method. Labels may be transferred into symbols now that SURF (Speeded Up Robust Features) features have been extracted from the pictures and histograms have been constructed. Classification is accomplished with the aid of Convolutional Neural Networks and the Support Vector Machine (SVM) (CNN). In addition, a pleasant graphical user interface is created for the client. According to [15], the ASL alphabets may be identified using convolutional neural networks. (CNNs). In [16], the author used a combination of k-nearest neighbor and support vector machine trained on generated characteristics from sensory input to categorize the 26 American Sign Language (ASL) equivalents of the English alphabet. The author's goal in [17], when he volunteered his skills, was to help ensure the company's continued prosperity. Using a hierarchical data architecture that takes into account user queries in light of current developments and demands, we provide a recommendation system for LCSs (RS-LCS). The company's approach includes operating under tight deadlines. A unique LCS structure based on group ratings and a collection of services is presented in this study [18]. Incorporating a group recommendation system into the LCS network, they seek to improve its responsiveness to new information. [19] Take user ratings and reviews into account by training a Neural Network to provide more accurate recommendations. The ability of a customer to repay a loan may be determined with great precision by optimizing a set of parameters, as shown in [20]. The primary goal was completed using machine learning methods and the Python language. A deep learning model built using tensor flow can determine whether or not a loan applicant is creditworthy by utilizing Min-Max standardization, Logistic Regression, a Random Forest classifier, and an image classification system.

3. Problem Statement

Over the last several years, researchers have developed a plethora of deep-based models with varied degrees of sophistication. A wide range of models might be presented if various types of data were combined with various types of input. Instead of relying on a static picture paradigm, which is typical of most signs, dynamic and continuous dynamic signs are used to overcome difficulties with video input. Movement epenthesis is complicated by continuous dynamic signs because the input video frequently shifts between different types of signs. Therefore, researchers have focused on identifying static and dynamic sign language rather than constantly dynamic sign language. There may be a need for techniques that reproduce spatio-temporal modeling of structures and patterns in current models if we're going to be able to accurately localize signals, gestures, and postures. These methods increase the model's complexity, but they improve its performance by adding in discriminative data. The use of deep learning's parallel computation capabilities, the implementation of more accurate fusion methods to integrate multiple input modalities (especially in continuous dynamic sign language), and the adaptation of traditional methods with deep-based models can all contribute to a reduction in the complexity of models used in sign language recognition and related areas.

4. Proposed Work

Deaf people have a visually appealing and practically significant problem when trying to communicate with hearing people who don't utilize sign language. The goal of this study is to determine whether the application of deep learning to the challenge of sign recognition is indeed possible. The full procedure of sign language recognition is shown in Figure 2.

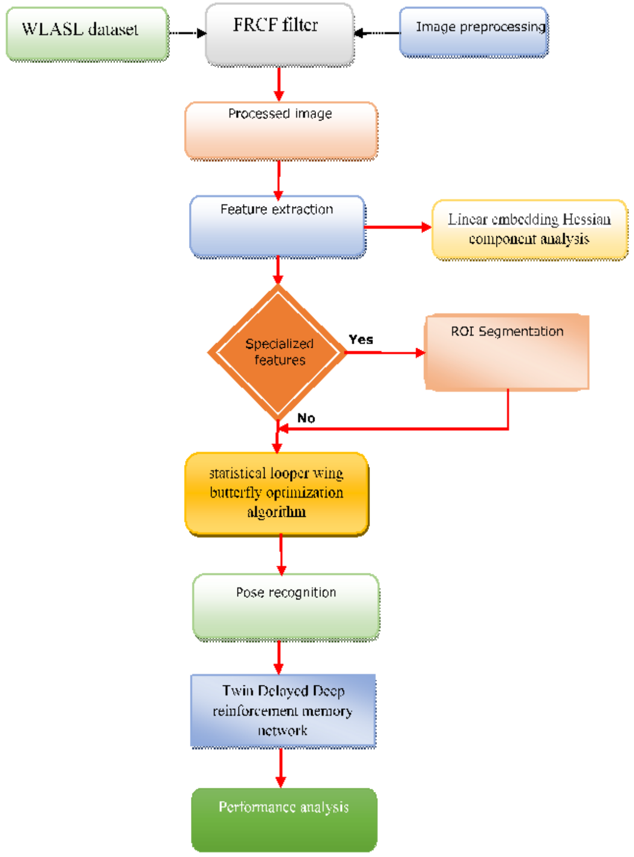


Fig. 2. Pictorial representation of the suggested architecture

1.1 Dataset

WLASL has more than 2,000 individual words, making it the largest video collection for Word-Level American Sign Language (ASL) recognition. Long-term, we expect that progress in studies of sign language comprehension, which we hope WLASL will encourage, will lead to better communication between the deaf and hearing populations.

1.2 Preprocessing

For the purpose of optimizing the AI tool's efficiency, preprocessing necessitates normalization, which encompasses any and all methods targeted at standardizing the input according to a set of established standards. This process, which may include a number of statistical procedures or media processing activities, is often carried out at the pre-processing stage of data collection. The input format (text, picture, or video), sample variability, machine learning architecture, automation tool goal, etc. all play a role in determining the normalization technique that is most suited to the present implementation. . Most state-of-the-art approaches to Sign Language Recognition include normalization because empirical evidence suggests it is effective. Due to the diversity of input modalities and study goals in SLR research, several normalization procedures have been developed and used. Many of the techniques are visual in nature, and they involve editing pictures in a certain way so that the algorithm can understand them more simply. The FRCF filter is used for this purpose; it approximates the pixel polymonial and corrects for the error during input processing. For this purpose, Chebyshev polynomials are used.

$P_0(y), P_1(y)$,... over the range [1,1] specified. The following equation is a representation of these concepts.,

$$P_1(y) = cos(karccosy), \quad y\epsilon[-1,1] \tag{1}$$

"The polynomials of the lowest order are stated clearly by: $P_0(y) = 1, P_1(y) = y, P_2(y) = 2y^2 - 1$, and so on. Using the recurrence relation, the higher-order polynomials may be calculated".

$$P_{k+1}(y) = 2yP_1(y) - P_{k-1}(y) \tag{2}$$

Because this is what we mean by "definition" (11). Use the coefficient b $_{(k,m)}$ to represent the relationship between y^m in $P_1(y)$, so that

$$P_1(y) = \sum_{m=0}^{k} b_{k,m}\, y^m \tag{3}$$

Remember that the coefficients (b $_{(k,m)}$) undergo a recursion as a result of (3). It may be used to reliably and quickly calculate (b $_{(k,m)}$). Coefficients may be calculated in advance and kept in a table for later use. As a consequence of (3), $P_1(y)$ has k zeros in the range [1,1] given by,

$$\xi_l = cos\left[\frac{\pi(2l-1)}{2k}\right] \quad (l = 1,2,..,k) \tag{4}$$

As such, the Chebyshev polynomials are said to fulfill a "discrete" orthogonality relationship.

$$\sum_{l=1}^{k} P_i(\xi_l)P_i(\xi_l) = \begin{cases} 0 & i \neq j, \\ {}^{1}/_{2} & i = j \neq 0, \\ k & i = j = 0. \end{cases} \tag{5}$$

A given function $g(y), y \in [-1,1]$ linear expansion approximation in terms of Chebyshev polynomials.

$$g(y) \approx \sum_{k=0}^{M} c_k\, P_k(y) \tag{6}$$

where the size of the expansion is determined by the coefficients (c_k). One should note that the function on the right-hand side is a polynomial of degree M. Zero-point sampling of the function g(y)$\{\xi_l: l = 1,..,M +$

$1\} \, of \; P_{M+1}(y)$ is a very straightforward method of fixing the coefficients (y). To continue, we use the orthogonality connection in (14) to our advantage and get,

$$c_k = \begin{cases} \dfrac{1}{M+1}\displaystyle\sum_{l=1}^{M+1} g(\xi_l), & k = 0 \\ \dfrac{2}{M+1}\displaystyle\sum_{l=1}^{M+1} g(\xi_l)\, P_k(\xi_l), & k \neq 0 \end{cases} \qquad (7)$$

The minimax estimate of g(y) is quite similar to the Gaussian approximation in (7) produced by using (6).

(4) was called the Gauss-Polynomial approximation because it is the sum of two Gaussians and a polynomial. The approximate fraction is found by plugging (6) into (7) and calculating its numerator and denominator. The numerator may be written as, if we remember that $P = e(i-j)$ and $\tau = e(i)$ then,

$$\textstyle\sum_{j\in\Omega} h\sigma_r(j)e(i-j)exp\left(-\frac{e(i)^2}{2\sigma_\gamma^2}\right) \qquad (8)$$

$$exp\left(-\frac{e(i-j)^2}{2\sigma_\gamma^2}\right)\textstyle\sum_{m=0}^{M} d_m \left(\frac{e(i)e(i-j)}{\sigma_\gamma^2}\right)^m \qquad (9)$$

“Next, for m = 0,...,M, we construct the images $h_m : I \rightarrow S$ and $E_m : I \rightarrow S$ given by”

$$h_m(i) = \left(\frac{e(i)}{\sigma_r}\right)^m \, and \; E_m(j) = exp\left(-\frac{e(i)^2}{2\sigma_r^2}\right) h_m(i)(10)$$

“We then consider the output of the filtering $\bar{E}_m: I \rightarrow \mathbb{S}$ of each E_m, given by”

$$\bar{E}_m(i) = \textstyle\sum_{j\in\Omega} h\sigma_r\,(j)E_m(i-j) \qquad (11)$$

Finally, the input images can be processed precisely.

4.3 Feature extraction

Feature extraction is crucial for SLR models because it affects how the models are trained and, ultimately, how quickly they may become proficient at distinguishing between different signs/words. Each characteristic is extracted from the raw data in the same fashion, and corresponds to a different part of the hands and face that is used in sign language. Traits are calculated statistically, with more weight given to those with higher discriminating power. Due to the vector representation of features in the latent space, the neural model can potentially learn the probability that a given feature is associated with a given class. Here, we use the LEHCA in order to extract the relevant details. Given a point p= (p, q) in the interest area, the Hessian matrix G (p,) in p at scale in Image J is defined as follows:

$$"G(P,\sigma) = \begin{bmatrix} Jpp(P,\sigma) & Jpq(P,\sigma) \\ Jp,q(P,\sigma) & Jq,q(P,\sigma)" \end{bmatrix} \qquad (12)$$

Where $Jpp, Jpq,$ and Jqq for every p in the picture, are the second derivatives. It explains how to examine the patterns of key areas for detecting sign language. Each feature's descriptor is computed using a method that leverages the best of both the Hessian and the non-Hessian features. To explain the LEHCA, it helps to think of it as a variant of the local binary pattern (LBP). In LEHCA, the primary formula is,

$$\text{LEHCA}(P_c.\,q_c) = \textstyle\sum_{m=0}^{m=3} R(Hm - Hm + 4)2^m \qquad (13)$$

To the extent that Hm is a group of eight adjacent pixels.The Hessian feature matrix characterizes the second-order fluctuation in local picture intensity around the chosen voxel..

$$"G(p,q) = \begin{bmatrix} K_{pp}(p,q) & K_{pq}(p,q)" \\ K_{qp}(p,q) & K_{qq}(p,q) \end{bmatrix} \qquad (14)$$

Where,

$$K_{pp}(p,q) = H_{pp}(p,q) * J(p,q)$$

$$K_{pq}(p,q) = H_{pq}(p,q) * J(p,q)$$

$$K_{qp}(p,q) = H_{qp}(p,q) * J(p,q)$$

$$K_{qq}(p,q) = H_{qq}(p,q) * J(p,q)$$

“where * is the feature operator, $J(p,q)$ is the processed image, and $H_{pp}(p,q)$, $H_{pq}(p,q)$, $H_{qp}(p,q)$ and $H_{qq}(p,q)$ are functions of the matrix G(p,q) to which the second derivative has been applied. For the matrix function, see equation 15”.

$$H(p,q) = \frac{1}{2\pi\sigma^2} exp\left(\frac{-p^2+q^2}{2\sigma^2}\right) \qquad (15)$$

Finally, the interested region related features are extracted.

4.4 Segmentation

Here, the, statistical looper wing butterfly optimization process may be used to zero in on the area of interest. Butterflies with elongated hindwings, or "loopers," fall within the genus Lepidoptera in the Linnaean order of the animal kingdom. More than 18,000 distinct butterfly species have been identified around the world. The sharpness of their senses is largely responsible for their incredible longevity. In order to find food and a mate, butterflies employ their senses of sight, smell, touch, taste, and hearing. Additional advantages of these senses include the ability to avoid detection by predators, move quickly from one area to another, and lay eggs in safe locations. Most importantly, butterflies can smell nectar, their primary food source, from great distances. Some pictures of the butterflies are shown in Figure 3. A major branch of butterfly optimization that takes its cues from nature is SLWBO. Butterflies are employed as optimization search elements in SLWBO because of their vital role in the process of finding food.

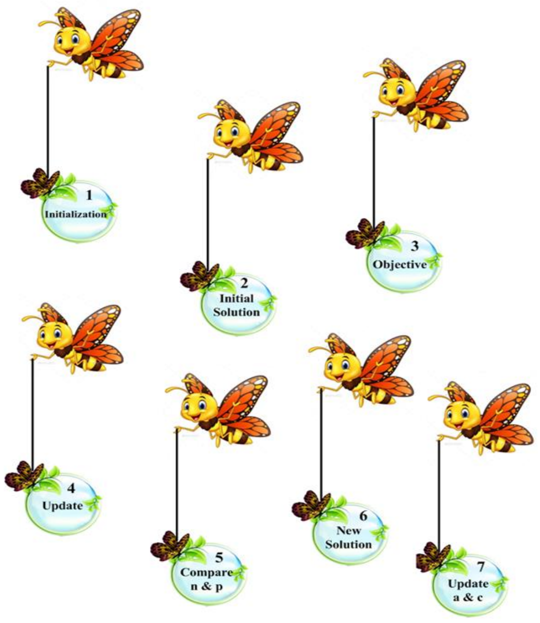


Fig. 3. Steps of optimization

SLWBO, proposed by S. Arora et al., is a population-based, biologically inspired optimization method. There was an SLWBO trend in 2018 around parodying buttery cuisine and mannerisms. In SLWBO, butterflies are said to emit a unique aroma associated with a certain kind of energy. This aroma is associated with the butterfly's fitness, as measured by the objective function. Every time a butterfly moves to a new location inside the mission space, its fitness level will increase. The butterfly's aroma may be detectable to other nearby butterfly species. A butterfly enters the SLWBO global search phase when it identifies the best-smelling butterfly in the search region and flies towards it. If a butterfly cannot identify another butterfly's scent in the search region, it will engage in what is known as the local search stage, in which it will roam about randomly. As demonstrated in Equation (above), in SLWBO, the intensity of the olfactory stimulus (the Interested area) is proportional to the physical strength of the stimulus (the hand's shape and orientation)

$$te_i = d \times J^b \qquad (16)$$

where te_i consist of the sensory modality (d), the stimulus intensity (J), and the strength exponent (b), which varies with modality and accounting for absorption degree (J). As can be seen in Equation (17), the optimization method may be utilized to relocate a digital butterfly in SLWBO.

$$Y_j^{p+1} = Y_j^p + E_j^{p+1} \qquad (17)$$

where Y_j^p is the jth butterfly's solution vector in the P-iteration sequence and E $_j$ is a description of the smell used by the y$_j$th butterfly to improve its position at every iteration. Moreover, the algorithm has two essential phases: local and global search. As shown in Equation, the butterfly makes progress toward the optimal solution h* during the global search phase (18)

$$E_j^{p+1} = (r^2 \times h^* - y_j^p) \times te_j \qquad (18)$$

In this case, h^* is superior than all other iteration solutions currently available. te_j indicates how jth butterfly interpreted a fragrance. One way to characterize the local search is using Equation (19):

$$E_j^{p+1} = (r^2 \times y_i^p - y_l^p) \times te_j \qquad (19)$$

where y_j^p and y_l^p are derived from the butterfly solutions in space. When both y_i^p and y_l^p when r is a random number between 0 and 1, then the butterflies a and b are members of are the same species. Therefore, Eq. (20) is a completely arbitrary expression. SLWBO employs a transfer probability t to make the transition from a global to a local search.

$$Y_j^{p+1} = Y_j^p + (r^2 \times h_{best} - y_j^p) \times e_j \qquad (20)$$

“Where the fragrance (e_j) can be formulated as follows”:

$$e_j = d \times J^b \qquad (21)$$

where d stands for the sense involved, e $_j$ for the intensity with which the smell is experienced, and a for the power exponent proportional to the extent to which the smell is taken in, The magnitude of the stimulus is denoted by J, while the absorption of a scent is represented by the power exponent a.

$$a(c) = b_s - (b_s - b_e) \times sin\left(\frac{\pi}{\mu} \times \left(\frac{p}{P_{max}}\right)^2\right) \qquad (22)$$

where as b_s and b_e represent the beginning and ending values of the parameter b, μ is the tuning parameter, and P max is the maximum number of iterations.

$$d_{p+1} = d_p + \left(\frac{0.025}{(d_p \times P_{max})}\right) \qquad (23)$$

where P_{max} Parameter d starts at 0.01 and the method runs for up to iterations.

$$U_j^{p+1} = WU_j^p + d_1 r_1 (tbest_j^p - y_j^p) + d_2 r_2 (tbest_j^p - y_j^p)(24)$$

$$Y_j^{p+1} = Y_j^p + U_j^{p+1} \qquad (25)$$

where U_p and U_{p+1} denote the jth particle's velocity at iteration p and p + 1, respectively. Assuming r1 and r2 are independent random variables, $d_1 = d_2 == 2$, (0, 1). The formula for determining w is (26).

$$W(p) = W^{max} - \frac{(W^{max} - W^{min}) \times P_j}{P_{max}} \qquad (26)$$

where W^{max} = 0.9, and W^{min} = 0.2, and P_{max} symbolizes the maximum number of possible repetitions. The two extremes of the continuous feature vector that represents the area of interest, Max and Min, are designated as such. Sign language prediction ROI may be defined using the optimization method.

4.5 Classification

When training the deep learning model, the hand motion is detected as a series of layers, making the proposed model sequential. The model mediates the flow of data between the input and output stages. In this scenario, the user speaks words that are translated into labels, and is then presented with signs (images from the database) that correspond to those labels in rapid succession. This network utilizes mathematical processing to decipher the symbolic meaning of the information in our image while keeping the locational context intact. By default, the TDDRMN classification strategy provides a function that, when given a sample, calculates the probabilities of all possible labels and returns those values.

$$\pi(b|r) = T\left(b_p = b \middle| r_p = r\right) \qquad (27)$$

As many occurrences as feasible in the training data sample must be accurately labeled by the classifier agent. Maximizing the classifier agent's rewards occurs when the agent correctly identifies a sample

b_p:

$$b_p = \sum_{l=0}^{\infty} \gamma^l s_p + l \qquad (28)$$

In reinforcement learning, the value of a state and action pair is determined by a function called the Q function.:

$$Q^{\pi}(r,b) = E_{\pi}[h_p|r_p = r, b_p = b] \qquad (29)$$

The twin delayed equation forms the basis for the Q function, which can be written as.:

$$Q^{\pi}(r,b) = E_{\pi}\left[r_p + \gamma Q^{\pi}\left(r_{p+1}, b_{p+1}\right)|r_p = r, b_p = b\right] \quad (30)$$

The optimal classification strategy π^* for TDDRMN is a greedy policy under the optimal Q* sign function, which allows the classifier agent to maximize the cumulative rewards.

$$\pi^*\,(b|r) = \begin{cases} 1, & if\ b = argmax_b Q^*(r,b) \\ 0, & else \end{cases} (31)$$

Substituting (30) into (31), the optimal Q^* function can be shown as:

$$Q^*(r,b) = E_{\pi}\left[r_p + \gamma \underset{b}{max} Q^*\left(r_{p+1}, b_{p+1}\right)|r_p = r, b_p = b\right] (32)$$

Q functions are a tabular representation of the finite-dimensional state space with low dimensionality. Interaction data (r,b,s,r') received from is stored in memory M . The agent picks a subset, B, of transitions from the whole network, M, at random, and continues to categorize it using the loss function in the following way.:

$$K(\theta_l) = \sum_{(r,b,s,r')\,\epsilon \mathrm{A}} (x - Q(r,b;\theta_l))^2 \qquad (33)$$

“The expression of x, where x is the target sign meaning estimate of the Q function, is”

$$x = \begin{cases} s, & terminal = True \\ s + \gamma max_{b'} Q(r', b'; \theta_{l-1}), & terminal = False \end{cases} \quad (34)$$

where r' represents the next state of r and b' represents the behavior of the agent in r'.

“The derivative for the loss function (34) with respect to angle θ is”:

$$\frac{\Delta K(\theta_l)}{\nabla \theta_l} = -2 \sum_{(r,b,s,r')\epsilon \mathrm{A}} \left(x - Q(r,b;\theta_l)\right) \frac{\nabla Q(r,b;\theta_l)}{\nabla \theta_l} \qquad (35)$$

Under the ideal Q* function derived by minimizing the loss function, the greedy policy (36) seeks to maximize the total reward (35). In this way, the most effective system for classifying TDDRMN has been implemented, and it is, $\pi^* : R \rightarrow B$.

Let's assume x+ and x- are the target values for Q in the r+ and r- samples, respectively. The desired levels of Q for both the positive and negative samples are as follows in (34) and (35).

$$x^+ = \begin{cases} (-1)^{1-J(b=k),} & terminal = True \\ (-1)^{1-J(b=k)} + \gamma max_{b'} Q(r', b'), & terminal = False \end{cases} (36)$$

$$x^- = \begin{cases} (+1)^{1-J(b=k)\leftthreetimes,} & terminal = True \\ (-1)^{1-J(b=k)} \leftthreetimes + \gamma max_{b'} Q(r', b'), & terminal = False \\ \end{cases}$$

(37)

where J(Y) is an pointer function.

“The Q network loss function k(l) should be rewritten as the product of the loss functions for the positive and negative classes, k + (θ_l) and k - (θ_l), respectively. This proof demonstrates the derivative of k + (θ_l) and k - (θ_l)”.

$$\frac{\nabla \mathrm{K}_+(\theta_l)}{\nabla \theta_l} = -2 \sum_{j=1}^{T} (x_j^+ - Q(r_j^+, b_j; \theta_l)) \frac{\nabla Q(r_j^+, b_j; \theta_l)}{\nabla \theta_l} (38)$$

$$\frac{\nabla -(\theta_l)}{\nabla \theta_l} = -2 \sum_{j=1}^{T} (x_j^+ + Q(r_j^+, b_j; \theta_l)) \frac{\nabla Q(r_j^+, b_j; \theta_l)}{\nabla \theta_l} (39)$$

T represents the total number of items in the set of positive interpretations, and M represents the total number of items in the set of negative interpretations.

By substituting (36) into (37), (38) into (39), and finally adding the derivative of k + (θ_l) and k - (θ_l).

$$\frac{\Delta K(\theta_l)}{\nabla \theta_l} = -2 \sum_{n=1}^{T+M} (1 - p_n) \gamma \underset{b_n'}{max} Q(r_n', b_n'; \theta_{l-1}$$

$$-Q(r_n, b_n; \theta_l)) \frac{\nabla Q(r_n, b_n; \theta_l)}{\nabla \theta_l}$$

$$-2\sum_{j=1}^{T}(-1)^{1-J(b_j=k_j)},\frac{\nabla Q(r_j,b_j;\theta_l)}{\nabla\theta_l}$$

$$-2\lambda\sum_{i=1}^{M}(-1)^{1-J(b_i=k_i)},\frac{\nabla Q(r_i,b_i;\theta_l)}{\nabla\theta_l}$$

where p_n=1 if terminal=True, otherwise p_n=0.

Finally, the meaning word according to the sign are allocated.

Algorithm: TDDRMN

```
"Input: Segmented output
Output: Sign classification
 Begin
For
        $\pi(b|r) = T(b_p = b|r_p = r)$
Generate the cumulative rewards according to Equation
End for
End for
   //target sign
      For
$x = \begin{cases} s, & terminal = True \\ s + \gamma max_{b'} Q(r', b'; \theta_{l-1}), & terminal = False \end{cases}$
     Update the loss function
      End for
        End for
$\frac{\nabla K_+(\theta_l)}{\nabla\theta_l} = -2\sum_{j=1}^{T}(x_j^+ - Q(r_j^+, b_j; \theta_l))\frac{\nabla Q(r_j^+, b_j; \theta_l)}{\nabla\theta_l}$
$\frac{\nabla K_+(\theta_l)}{\nabla\theta_l} = -2\sum_{j=1}^{T}(x_j^+ - Q(r_j^+, b_j; \theta_l))\frac{\nabla Q(r_j^+, b_j; \theta_l)}{\nabla\theta_l}$
Sign mean {positive, negative}
End for
 End"
```

5. Performance Analysis

The proposed model for sign language identification is tested through a series of tests. The main purpose of the experimentation series is to evaluate the proposed TDDRMN model by applying it to a dataset. The majority of the testing was performed in a MATLAB environment.

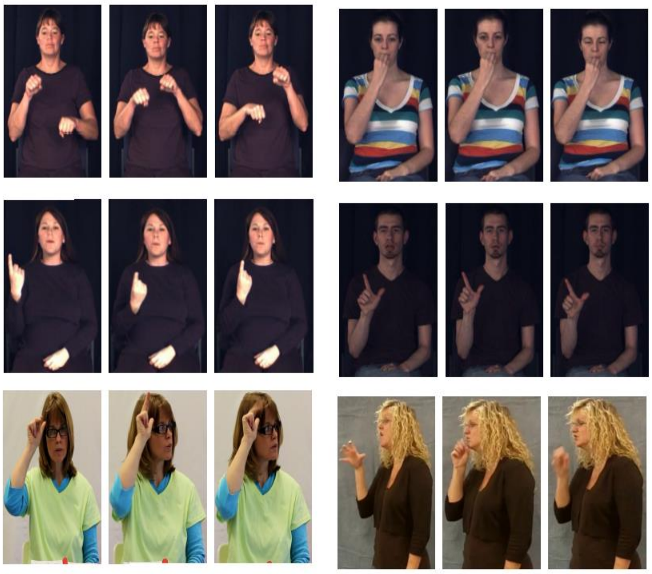
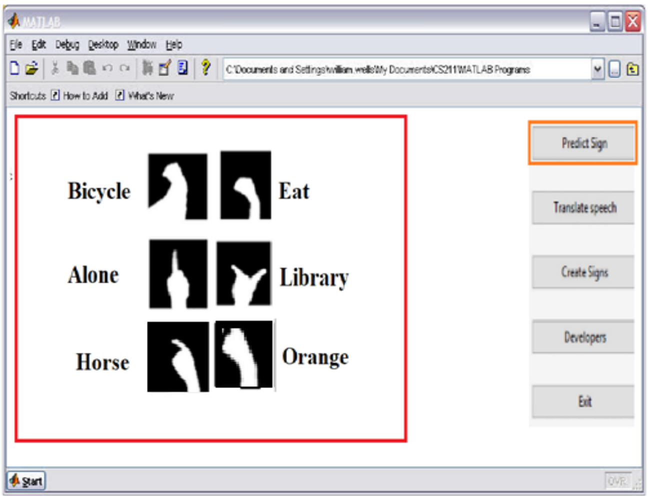


Fig. 4. Sample input

The sample input was depicted in figure 4.



MATLAB
File Edit Debug Desktop Window Help
C:\Documents and Settings\william.wells\My Documents\CS211\MATLAB Programs
Shortcuts How to Add What's New
Bicycle
Eat
Alone
Library
Horse
Orange
Predict Sign
Translate speech
Create Signs
Developers
Exit
Start

Fig. 5. Simulated output

The overall simulated output for the suggested classifier was depicted in figure 5.

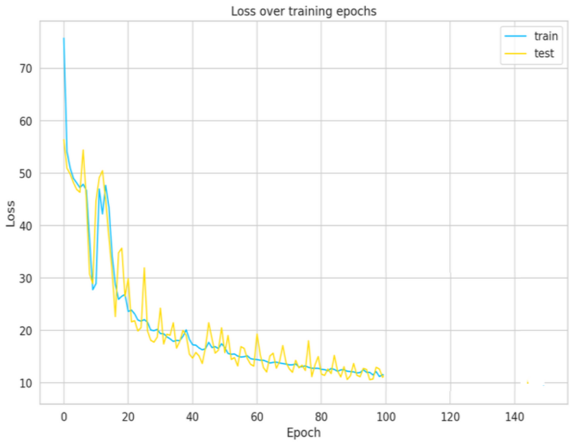


Fig. 6. Epoch Vs. Loss

An epoch's worth of data is used to calculate a loss function, which provides a quantitative measure of loss for the whole period. When developing an iterative curve, some data will always be lost. The resulting curve indicates that the costs associated with classifier training, validation, and testing are quite low when compared to those of competing approaches. If there is only a little discrepancy between the training loss and the validation loss, our model is probably underfitted. A larger sample might potentially decrease the amount of loss experienced during training (either the number of layers or the raw number of neurons in each layer). You can see the raw data used to determine the loss in Figure 6. Conversely, the loss statistic evaluates a model's performance on the validation set. Some of the information has been kept aside as a "validation set" for the purpose of verifying the quality of our model. The testing loss is equal to the total of the mistakes committed on the training set and the validation set. There is much less level loss with the suggested strategy compared to the alternatives.

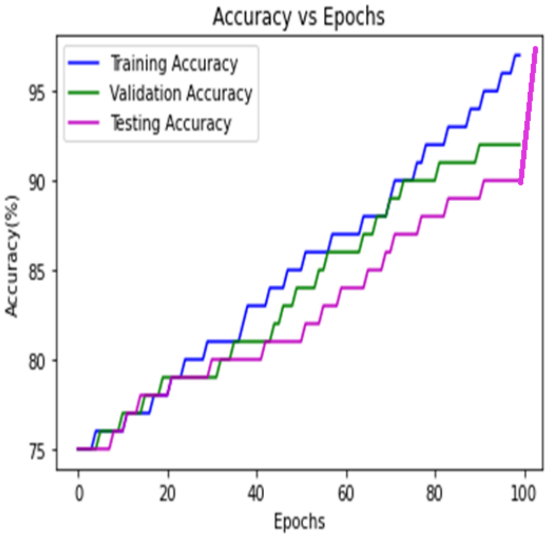
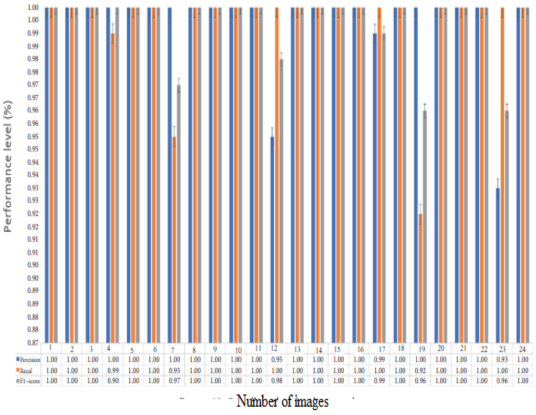


Fig. 7. Epochs Vs. Accuracy

As can be seen in Figure 7, both the prediction accuracy and the loss value rise with the number of epochs used. With 100 iterations, a deep learning-based TDDRMN algorithm outperforms test accuracy by 0.01%. Several metrics of performance may be compared to existing systems to prove the worth of the proposed method.

Table 1. Performance metrics

“Metrics”	**“Description”**	**“Formula”**
“Accuracy”	“The percentage of times that forecasts were accurate over the whole sample.”	“Accuracy = $\frac{RO+RB}{RO+DO+RB+DB}$”
“Recall”	“The rate at which instances are correctly predicted as a fraction of total images.”	“Recall = $\frac{RO}{RO+DB}$ "
“Precision”	“The rate at which instances are correctly predicted as a fraction of total images.”	“Precision = $\frac{RO}{RO+DO}$”
“F measure or F1 score”	“Only two measurements are needed to determine the F-measure (precision and recall).”	“F measure = $2* \frac{(Precision *Recall)}{(Precision+Recall)}$”



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
■ Percision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.95	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	0.93	1.00
■ Recal	1.00	1.00	1.00	0.99	1.00	1.00	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.92	1.00	1.00	1.00	1.00	1.00
■ F1-score	1.00	1.00	1.00	0.90	1.00	1.00	0.97	1.00	1.00	1.00	1.00	0.98	1.00	1.00	1.00	1.00	0.99	1.00	0.96	1.00	1.00	1.00	0.96	1.00

Fig. 8. Number of images Vs. Performance level

Figure 8 depicts the overall efficacy of the proposed mechanism. The effectiveness of the proposed method can be demonstrated by contrasting it with current approaches. [14,15]

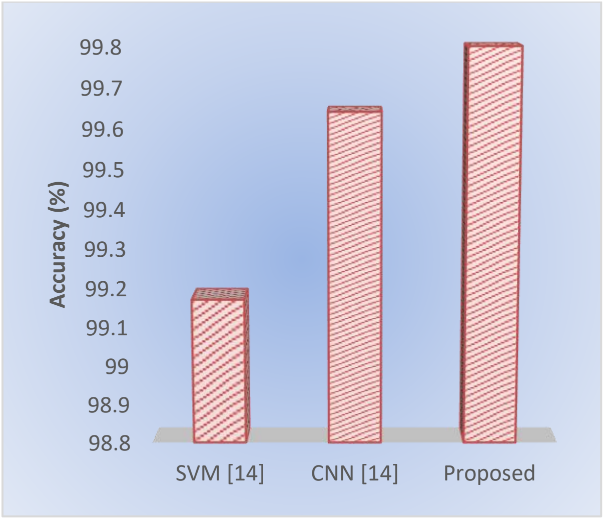


Fig. 9. Accuracy percentile calculation

When the TDDRMN model was put into place, the kind of sign could be determined with great accuracy. The proposed technique achieves higher levels of accuracy (99.8%), which is higher than the state-of-the-art mechanisms now in use.

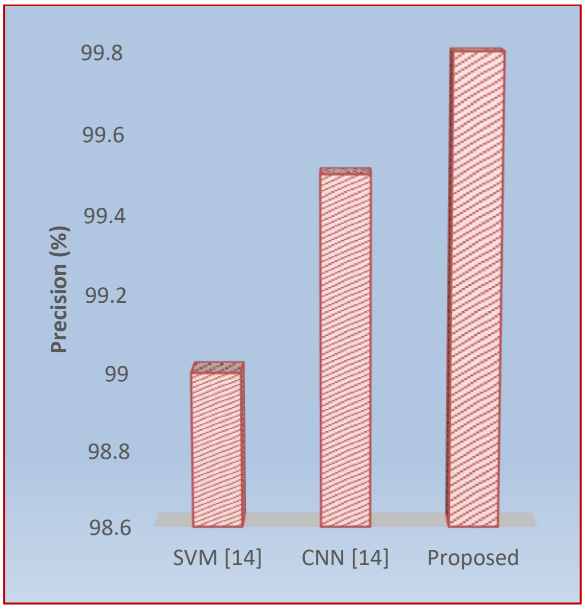


Fig. 10. Precision percentile calculation

As of from the segmented data the sign data's was classified. From the result obtained from the figure 10 the suggested methodology acquires high range of precision (99.8) which is very high when compared to other existing mechanisms in use.

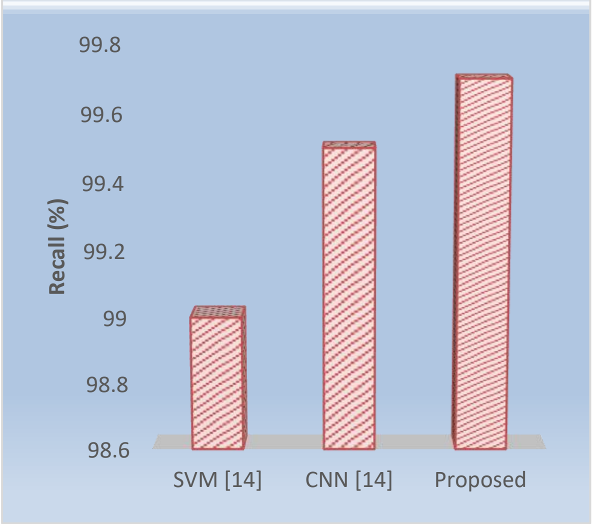


Fig. 11. Recall percentile calculation

Sign language detection recall findings for the WLASL datasets are shown in Figure 11. Once again, the recommended approach is the most effective of all the strategies considered by obtaining the satisfied level of recall (99.7%)

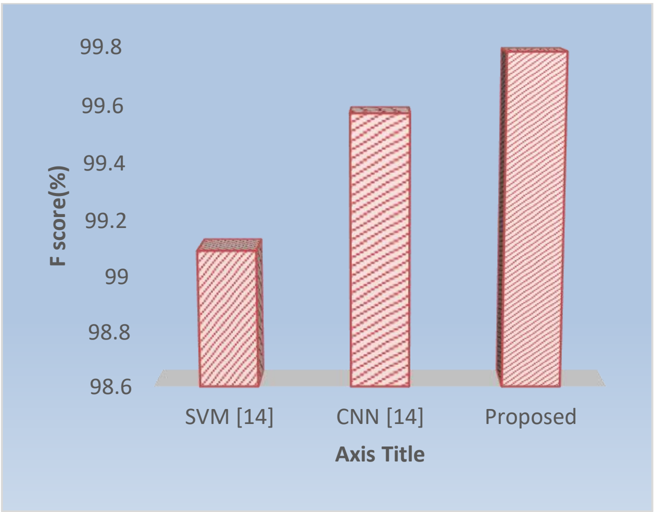


Fig. 12. F score percentile calculation

From the figure 12 it will be shown that the suggested methodology has high F score (99.7%) range when compared to other existing mechanisms

Table 2. Comparative performance analysis

Reference	Gestures analysis	Recognition accuracy (%)
[16]	"Alphabets"	79.8
[21]	"Numbers"	91.3
[22]	"10 selected gestures"	83.36
[23]	"26 ASL gestures (A–Z) and 36 ASL gestures (A–Z, 0–9)"	93.8
[24]	"30 ASL gestures (12 dynamic signs and 18 static signs)"	96.4
[25]	"24 ASL gestures"	99.6
Proposed	"Anything"	

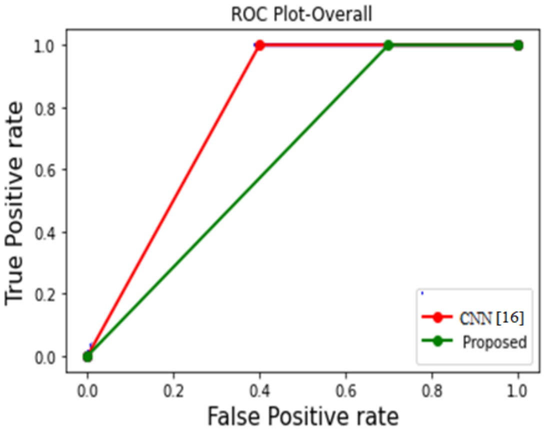


Fig. 13. False positive rate Vs. True positive rate

The global performance metrics offered in Figure 13 are included in the strategy there. Figure 13 shows ROC with a 0.99 value. The classifier properly identified the sign event if the area under the curve (ROC) score. A pair of sensitivity and specificity values are represented by each point on the ROC curve, which may be used to create cutoffs for decision-making. The capacity of a parameter to distinguish between various kinds of activity is measured using the area under the ROC curve. The outcomes unmistakably demonstrate that the suggested technique is better to the present one.

6. Conclusion

In this study, we provide a computer-vision-based, real-time, and sign-independent Sign Language recognition system. Each of the gathered images is run through the suggested segmentor to determine where in the frame a hand is most likely to be seen. Binary images of hand forms in sign language are recovered using an effective segmentation method and then used for either training or assessment. At the end of the day, the TDDRMN classifier is used to understand both vowel and consonant signals in Sign Language. This system achieved a computational accuracy of 99.8 percent overall. Voice recognition is the future of all computing interfaces, however those with speech or hearing impairments may have a more difficult time adjusting to this future. We plan on creating a Sign Language interface that can be used with popular voice assistants like Siri, Google Assistant, Alexa, etc. to demonstrate potential applications of the technology in the real world. Using the system-predicted sign language, the interface then translates the instructions into spoken voice instructions for the Assistant. Those who utilize sign language may thus benefit from a voice assistant

References

[1] Z. Zeng and F. Wang, "An Attention Based Chinese Sign Language Recognition Method Using sEMG Signal," in *2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, 2022, pp. 457-461.

[2] A. Mino, M. Popa, and A. Briassouli, "The Effect of Spatial and Temporal Occlusion on Word Level Sign Language Recognition," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 2686-2690.

[3] A. Elboushaki, R. Hannane, K. Afdel, and L. Koutti, "MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences," *Expert Systems with Applications,* vol. 139, p. 112829, 2020.

[4] K. M. Sagayam, A. D. Andrushia, A. Ghosh, O. Deperlioglu, and A. A. Elngar, "Recognition of hand gesture image using deep convolutional neural network," *International Journal of Image and Graphics,* vol. 22, p. 2140008, 2022.

[5] A. Dadashzadeh, A. T. Targhi, M. Tahmasbi, and M. Mirmehdi, "HGR-Net: a fusion network for hand gesture segmentation and recognition," *IET Computer Vision,* vol. 13, pp. 700-707, 2019.

[6] Y. Liao, P. Xiong, W. Min, W. Min, and J. Lu, "Dynamic sign language recognition based on video sequence with BLSTM-3D residual networks," *IEEE Access,* vol. 7, pp. 38044-38054, 2019.

[7] X. Ji, Q. Zhao, J. Cheng, and C. Ma, "Exploiting spatio-temporal representation for 3D human action recognition from depth map sequences," *Knowledge-Based Systems,* vol. 227, p. 107040, 2021.

[8] P. Wang, W. Li, Z. Gao, C. Tang, and P. O. Ogunbona, "Depth pooling based large-scale 3-d action recognition with convolutional neural networks," *IEEE Transactions on Multimedia,* vol. 20, pp. 1051-1061, 2018.

[9] L. Meng and R. Li, "An attention-enhanced multi-scale and dual sign language recognition network based on a graph convolution network," *Sensors,* vol. 21, p. 1120, 2021.

[10] V. A. Shanthakumar, C. Peng, J. Hansberger, L. Cao, S. Meacham, and V. Blakely, "Design and evaluation of a hand gesture recognition approach for real-time interactions," *Multimedia Tools and Applications,* vol. 79, pp. 17707-17730, 2020.

[11] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications,* vol. 164, p. 113794, 2021.

[12] R. Rastgoo, K. Kiani, and S. Escalera, "Multi-modal deep hand sign language recognition in still images using restricted Boltzmann machine," *Entropy,* vol. 20, p. 809, 2018.

[13] R. Rastgoo, K. Kiani, and S. Escalera, "Video-based isolated hand sign language recognition using a deep cascaded model," *Multimedia Tools and Applications,* vol. 79, pp. 22965-22987, 2020.

[14] S. Katoch, V. Singh, and U. S. Tiwary, "Indian Sign Language recognition system using SURF with SVM and CNN," *Array,* vol. 14, p. 100141, 2022.

[15] M. Kumar, P. Gupta, R. K. Jha, A. Bhatia, K. Jha, and B. K. Shah, "Sign Language Alphabet Recognition Using Convolution Neural Network," in *2021 5th International Conference on Intelligent*

Computing and Control Systems (ICICCS), 2021, pp. 1859-1865.

[16] C.-H. Chuan, E. Regina, and C. Guardino, "American sign language recognition using leap motion sensor," in *2014 13th International Conference on Machine Learning and Applications*, 2014, pp. 541-544.

[17] P. Kirubanantham and G. Vijayakumar, "Novel recommendation system based on long-term composition for adaptive web services," *Computational Intelligence,* vol. 36, pp. 1063-1077, 2020.

[18] P. Kirubanantham, S. Sankar, C. Amuthadevi, M. Baskar, M. Senthil Raja, and P. Karthik, "An intelligent web service group-based recommendation system for long-term composition," *The Journal of Supercomputing,* vol. 78, pp. 1944-1960, 2022.

[19] P. Kirubanantham, A. Saranya, and D. S. Kumar, "Convolutional Recommended Neural Network system based on user reviews for movies," in *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, 2021, pp. 17-21.

[20] P. Kirubanantham, A. Saranya, and D. S. Kumar, "Credit Sanction Forecasting," in *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, 2021, pp. 155-159.

[21] B. Khelil, H. Amiri, T. Chen, F. Kammüller, I. Nemli, and C. Probst, "Hand gesture recognition using leap motion controller for recognition of arabic sign language," in *3rd International conference ACECS*, 2016, pp. 233-238.

[22] T.-W. Chong and B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors,* vol. 18, p. 3554, 2018.

[23] W. Zehra, A. R. Javed, Z. Jalil, H. U. Khan, and T. R. Gadekallu, "Cross corpus multi-lingual speech emotion recognition using ensemble learning," *Complex & Intelligent Systems,* vol. 7, pp. 1845-1854, 2021.

[24] Y. Du, S. Liu, L. Feng, M. Chen, and J. Wu, "Hand gesture recognition with leap motion," *arXiv preprint arXiv:1711.04293,* 2017.

[25] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures," *IEEE Transactions on Multimedia,* vol. 21, pp. 234-245, 2018.