

Article

Progression Learning Convolution Neural Model-Based Sign Language Recognition Using Wearable Glove Devices

Yijuan Liang ^{1,2} , Chaiyan Jettanasen ¹  and Pathomthat Chiradeja ^{3,*}

¹ School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand; 63601009@kmitl.ac.th (Y.L.); chaiyan.je@kmitl.ac.th (C.J.)

² Guangxi Electrical Polytechnic Institute, Nanning 530299, China

³ Faculty of Engineering, Srinakharinwirot University, Bangkok 10110, Thailand

* Correspondence: pathomthat@swu.ac.th

Abstract: Communication among hard-of-hearing individuals presents challenges, and to facilitate communication, sign language is preferred. Many people in the deaf and hard-of-hearing communities struggle to understand sign language due to their lack of sign-mode knowledge. Contemporary researchers utilize glove and vision-based approaches to capture hand movement and analyze communication; most researchers use vision-based techniques to identify disabled people's communication because the glove-based approach causes individuals to feel uncomfortable. However, the glove solution successfully identifies motion and hand dexterity, even though it only recognizes the numbers, words, and letters being communicated, failing to identify sentences. Therefore, artificial intelligence (AI) is integrated with the sign language prediction system to identify disabled people's sentence-based communication. Here, wearable glove-related sign language information is utilized to analyze the recognition system's efficiency. The collected inputs are processed using progression learning deep convolutional neural networks (PLD-CNNs). The technique known as progression learning processes sentences by dividing them into words, creating a training dataset. The model assists in efforts to understand sign language sentences. A memetic optimization algorithm is used to calibrate network performance, minimizing recognition optimization problems. This process maximizes convergence speed and reduces translation difficulties, enhancing the overall learning process. The created system is developed using the MATLAB (R2021b) tool, and its proficiency is evaluated using performance metrics. The experimental findings illustrate that the proposed system works by recognizing sign language movements with excellent precision, recall, accuracy, and F1 scores, rendering it a powerful tool in the detection of gestures in general and sign-based sentences in particular.

Keywords: deep convolution neural networks; glove method; memetic optimization; sign language; progression learning; wearable devices



Citation: Liang, Y.; Jettanasen, C.; Chiradeja, P. Progression Learning Convolution Neural Model-Based Sign Language Recognition Using Wearable Glove Devices. *Computation* **2024**, *12*, 72. <https://doi.org/10.3390/computation12040072>

Academic Editor: Yudong Zhang

Received: 5 February 2024

Revised: 24 March 2024

Accepted: 25 March 2024

Published: 3 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Sign language, a structured sequence of hand motions, provides the most efficient means of communication for deaf people. Hard-of-hearing people have traditionally used sign language to interact with each other. The available approaches to the interpretation of sign language utilize vision techniques to translate signs autonomously. This primarily entails two tasks: isolated or word-based recognition and continuous or sentence-based recognition. The configuration of the fingers, the direction of the hand's motion, and the placement of the hand in relation to the body serve as the fundamental components of arranged communications in sign language. Due to the prominence of autonomous devices like glove sensors and the rapid increase in people considered deaf, the detection of hand gestures has become more important. Using AI methods, it is possible to improve the capacity for decision making through the application of electronic technologies. The

smart-glove wearable device is designed to help users convert sign language motions into spoken or written sentences, facilitating communication.

Progression learning entails training on both the various gesture activities exhibited during the movement of wearable glove devices and data subsets to gain knowledge gradually.

Progression learning is more about preserving and growing information as time passes than transferring understanding from one activity to another. The purpose of progression learning for network performance optimization is to identify the best neural architecture to use for the development of an individual image classification task, especially for sign language recognition.

Generally, the recognition of sign language is challenging owing to its learning intricacy [1] and the variable techniques used to capture motion, direction, and hand positions. People frequently use hand gestures to express their thoughts and feelings; hence, research has focused on hand gestures in sign language prediction [2]. Conversely, dynamic gestures address the issue of isolating hand-gesture identification in RGB films and can either be single words or continuous sentences. Traditional CNNs are ineffective in learning the representation of signs [3]. The pertinent patterns are only present in the motion of the field of the signer's hands, although there are many irrelevant patterns in the scene as a whole. Hence, it is necessary to focus on static and dynamic types by employing Python, OpenCV, and intelligent gloves [4] that can recognize sign language. Markov properties [5] can be seen in the problem of the dynamic recognition of gestures. This issue arises when the conditional likelihood of the density of this moment's event, given all previous and present occurrences, depends solely on the recent event. This problem makes use of techniques like time-compressing templates. For individuals in these random populations of disabled people, communication hurdles arise from factors such as sign language variance [6], sensor setup, classifying signs, study design, and the inability to comprehend or employ sign language. A continuous or dynamic sentence in sign language is broken into subwords via data segmentation [7], a crucial preprocessing operation. This operation is performed by detecting the hand's structure, orientation, and motion. These are then modeled further, and the associated component classifiers are taught. Since they can measure finger movements, sensory gloves [8] are useful in a wide range of applications, particularly those required for real-time data collection. Vision and wearable glove devices are the most useful approaches for understanding sign language [9]. Numerous academics have suggested systems for recognizing sign language based on these methods. The glove-based system employs electromechanical or optical indicators attached to the signer's glove. It then transforms the movement of the fingers into electronic signals [10] to determine the hand position for the recognition of messages being communicated. Each gesture is classified using a deep learning algorithm, based on CNNs [11], that continuously learns and extracts the characteristics of each hand movement. The CNN model, employing the Kinect source of collected information, is used to extract depth motion trajectory information and perform spatiotemporal analysis of the joint features of the body [12]. Then, a cutting-edge data glove with two forearm rings and a designed three-dimensional in-nature flex sensor [13] is used to record the motion of the whole arm and all fingers in fine detail. Rather than using a word model, the Recurrent Neural Networks (RNNs) developed for subunits [14], employing factorization concepts for sign language recognition, are preferred in the extraction of hand motion features, showing maximum accuracy. The ongoing improvement of AI is a long-standing development goal of learning, wherein agents learn from, keep, or store tasks performed in order to apply knowledge from earlier tasks to speed up convergence [15]. Progression learning gradually raises the training workload for the translation of sign languages by expanding among its subnetworks according to a growth schedule, represented by a series of smaller networks like subwords, with increasing dimensions observed for all training epochs [16]. It is specifically used to make sure the network is adequately optimized after each growth.

The current key contributions include the following:

- Developing a sign language recognition system capable of accurately interpreting sentence-based sign language identification;
- Promoting accessibility for hearing-disabled individuals by improving communication through AI-driven solutions;
- Employing word error rate recognition and improving the accuracy of hand gestures using the progression learning capability of wearable glove sensors.

Section 2 of this paper focuses on the sign language prediction models used. In Section 3, this article then describes the learning progress of the PLD-CNN model, including the memetic optimization technique for improving prediction accuracy. The results and evaluation metrics are investigated in Section 4. Finally, a discussion and future directions are given in Section 5.

2. Literature Review

Rastgoo et al. (2022) [17] applied a combination of three deep convolutional neural networks (3DCNNs) to a hand sign language approach in order to estimate the specified angles of 3D key hand points between fingers, with Singular Value Decomposition (SVD) used for the feature extraction of key points from hand estimators in a real-time scenario. This was followed by the use of Long Short-Term Memory (LSTM) to identify the word sign from the fully convoluted layer, alongside the use of the Adam optimizer to enhance the network parameters. This overall technique had a recognition time of words of a minimum of 2 s and an accuracy of 99%. Since it uses a real-time scenario, the approach faces computational complexity due to the operations.

Al-Hammadi et al. (2020) [18] proposed a dynamic recognition model to generate both signer-dependent and signer-independent methods using two modes of a 3D deep convolutional neural network (3DCNN + MLP + En) with a multi-layer perceptron and an encoder for feature fusion. This study was optimized using a mini-batch gradient descent approach in order to enhance the network parameters. The maximum recognition accuracy achieved was 98%, and the input clip restriction was the use of only 16 frames of gesture space.

Aly and Aly (2020) [19] employed Hybridized Deep Learning models to generate a signer-independent Arabic Sign Language Prediction (HDL-ASLP) model. This model included the DeepLabV3+ for hand segmentation and a convolutional self-organizing map for feature extraction, making predictions using deep Bidirectional Long Short-Term Memory (Bi-LSTM). These approaches achieved 89.59% accuracy and a Mean Intersection of Union (MIOU) of 94.2 when applied to 23 words from three signers performing Arabic sign language gestures. This study was optimized with a stochastic gradient descent approach to enhance the network parameters. This was performed using the King Saud University Saudi-based Sign Language (KSUS2L) data source, and the only research gap identified was the limited availability of data samples for recognition.

In order to identify spatial and temporal patterns in human pose space, Li et al. (2020) [20] implemented two types of deep learning models based on pose and appearance (pa) using Temporal Graph CN (pa-TGCN). The evaluation impacted the size of vocabulary and spatial features, and the Adam optimizer was used to train input sample parameters. Word-level American Sign Language (ASL) data sources with 2000 words from 100 signers were chosen, and an accuracy of up to 62.63% was obtained with few-shot learning methods.

DelPreto et al. (2022) [21] developed a pre-trained LSTM model to detect static and dynamic gestures from an ASL data source of 12 words. The network was frequently responsive to slight positional changes, particularly when fingers were in contact, which resulted in significant resistance changes. Additionally, there was potential for some prediction delays to occur when the knit's prior recorded response times were combined with the 2 s categorization window. The wearable device comprised an intelligent glove with an accelerometer fitted to track mobility and strain-sensitive resistances obtained to capture postural data. The Root-Mean-Square Error (RMSE) was 0.010, and there was an

accuracy of 99% when calculations used the average means and standard deviations of the identified true gesture levels.

Rosero-Montalvo et al. (2018) [22] presented an intelligent wearable glove device for the sign language translation of numbers using an evolutionary type of cross-generational elitist selection, heterogeneous recombination, and a cataclysmic mutation (CHC) algorithm with five kinds of flex sensors for data collection. This was balanced using the Kennard–Stone method, followed by the classification of signs using the K Nearest Neighbor (KNN) technique. The training set produced a classification accuracy of 87.8%, with a minimum of 64% of the information being stored using five colors labeled 1 to 5: black, red, green, blue, and cyan. The limitation of this procedure was its exclusive focus on translating numbers.

Zhang et al. (2019) [23] suggested a multimodal CNN, using LSTM to model temporal dependencies using the MyoSign model, and then detached models from the inputs of many sensory channels. Then, Connectionist Temporal Classification (CTC) was used to overcome temporal gaps and perform continuous analysis throughout the entirety of sign language recognition from ASL at both word and sentence levels. The authors obtained an accuracy of 93.7% at the word level and 93.1% at the sentence level, evaluating the run time of both words and sentences and identifying the error sequences in the possible combinations. The fact that, when people are walking, the inertial signal pulses may be altered by body motions is a major drawback of this design.

Nandi et al. (2023) [24] suggested using CNNs featuring multiple gradient optimization methods for sign language recognition, aiming to eliminate the deaf–dumb–normal barrier. A CNN was used to process 62,400 images from 26 participants, with data augmentation used to remove extraneous information and resize features. The use of batch processing and a dropout layer reduced image redundancy after refining the network parameters with a diffGrad optimizer. The sign language recognition system brought greater improvements than other approaches. The sign language recognition system was constructed using machine learning, computational image processing, and optimization methods. The study then investigated gestures in several directions in order to improve detection accuracy, focusing only on static signs from Indian sign language and failing to focus on dynamic gestures.

In 2022, Rwelli et al. [25] recommended using wearable sensors to recognize Arabic sign language gestures based on a machine learning technique called a deep convolutional network (DCN). The purpose of doing so was to enable the extraction of features from sensor devices worn by hearing-impaired people. The study investigated 30 hand sign letters captured from gloves and recognized around 90% of the samples. The experimental procedure was used to acquire results for various training samples and detect sign language based on hand gestures. The limitations of this study included its size and exclusive focus on individual letters.

Table 1 summarizes the existing research by various scholars: Numerous studies using AI have been conducted to address various sign language recognition challenges for different languages. The existing limitations, like the shortage of studies into inertial signal pulses, need to be considered when designing the proposed algorithms to replace existing languages like HDL-ASLP [19], pa-TGCN [20], and CNN-LSTM [23]. The findings of this literature review may assist in efforts to create reliable and signer-centered wearable glove-based sign language recognition systems. The three existing models chosen for comparison purposes involve input systems with data collected from video datasets and the use of wearable devices with inertial and electromyographic signals to capture signs.

The reviewed studies were highly relevant to sign language recognition, with recent advancements in deep learning-related works including the release of wearable technology like 3DCNN [17], 3DCNN + MLP + En [18], pa-TGCN [20], LSTM [21], CNN-LSTM [23], CNN-GO [24], and DCN [25].

Table 1. Summarization of literature studies regarding sign language recognition.

Ref. No.	Algorithm Used	Data Source	Samples/Feature	Metrics Evaluated
[17]	3DCNN	1. RKS-PERSIANSIGN; 2. First-Person; 3. American Sign Language Lexicon Video; 4. isoGD	RGB video to word labels	Accuracy—99%; recognition time—2 s
[18]	3DCNN + MLP + En	(KSUS2L)	40 persons	Accuracy—98.75%
[19]	HDL-ASLP	Arabic sign language database	23 words	Accuracy—89.59%; MioU—94.2%
[20]	pa-TGCN	ASL	2000 words	Acc—62.63%
[21]	LSTM	ASL	12 letters and 12 words	Accuracy—2.8%; prediction time of 0.08 s
[22]	CHC	Sensor data	Numbers 0–9 (10 signers)	Accuracy—87.8%; execution time—30 min
[23]	CNN-LSTM	ASL	70 words, 100 sentences	Accuracy—93.7% and 93.1%
[24]	CNN-GO	Indian sign language	62,400 images	Accuracy—99%
[25]	DCN	Arabic sign language	30 hand sign letters	Accuracy—90%

3. Proposed System

Sign languages, each possessing distinctive linguistic patterns, are the principal means of communication for the deaf in society. Smart sensor innovations, integrated with AI and Machine Learning (ML), make wearable systems more powerful. The main goal of this research is to develop a wearable device-based sign language recognition system that uses deep convolutional neural networks to enable real-time interpretation and translation. The technical features of gesture recognition and sign-language-specific syntactic rules are highlighted using progression learning.

Glove-based systems can significantly influence numerous application fields because hand gestures are capable of conveying especially important data. Each gesture is classified using a deep learning algorithm, based on CNN, that continuously learns and extracts characteristics, employing a progression learning technique for this continuous learning process. A progression learning algorithm can be trained on this sign knowledge and correlate a certain hand gesture with its associated sign by gathering a moderate amount of information for every single word, divided and separated from the sentences, to provide an input. Progression networks are a step in the right direction since they resist forgetting and may use existing knowledge by making indirect links to previously learned elements, such as words and characters. Real-time hand gesture recognition is one of the most difficult study areas in human–computer interaction. Progression networks are an initial step towards developing a fully continuous learning agent because they have the components needed to learn several tasks sequentially while also facilitating transfer and being resistant to catastrophic forgetting.

The system should recognize sign language sentences when words are in a syntax-specific order. This means that the proposed PLD-CNN system analyzes signals, words, and the arrangement of sentences and language. The algorithm learns to recognize full phrases by interpreting sequential sign language patterns and structures through iterative improvement, improving hearing-impaired communication.

A MATLAB program was developed to prepare the inertial measurement unit (IMU) data. The IMU data are first preprocessed using a moving average filter, and then the smoothed IMU data are saved in separate CSV files. Sign languages differ by place, but they share fundamental gestures. This research develops a glove-input sign language recognition system for people in the deaf and hard-of-hearing communities. Training is

performed on datasets containing several sign languages in order to address sign language variety. PLD-CNNs and HMMs help the system comprehend the rules of grammar and recognize certain gestures in order to establish a global sign language recognition system.

Figure 1 depicts the overall implementation procedure of the PLD-CNN algorithm for sign language prediction.

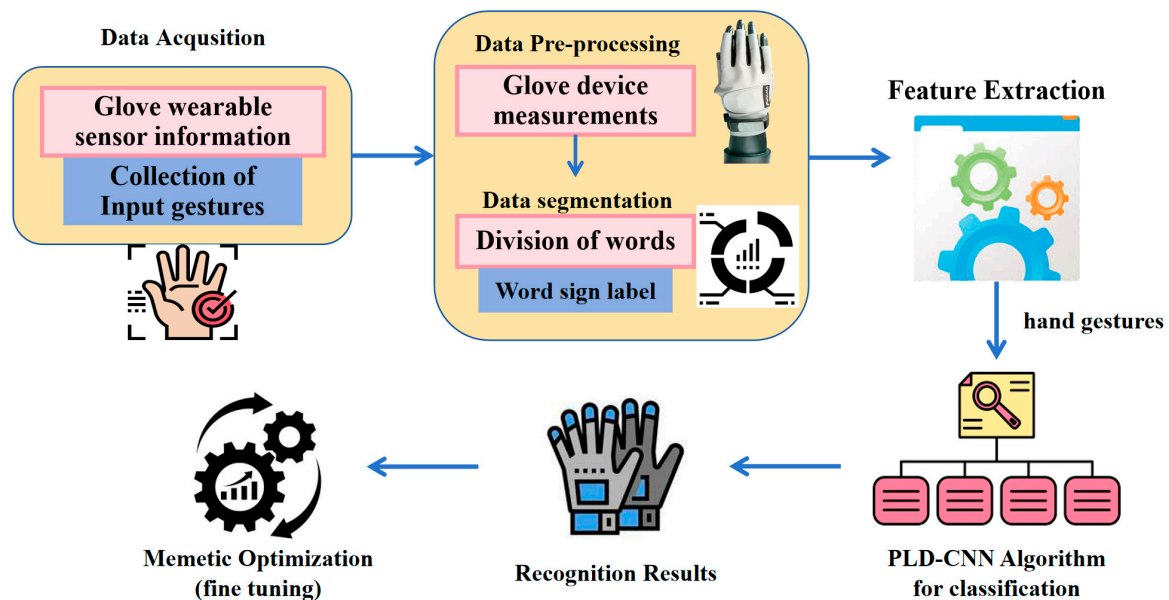


Figure 1. The proposed framework for PLD-CNN-based sign language recognition.

Figure 1 represents the structure of PLD-CNN-based sign language recognition. The system consists of several phases, such as data acquisition, preprocessing, feature extraction, sign language classification, and efficiency analysis. Initially, the data are collected from wearable glove sensors, which consist of several pieces of information. The gathered input gestures are preprocessed by examining the device measurements. Then, words are divided, and sign labels are segmented. The segmented details are fed into the feature extraction stage. The extracted features are processed using the PLD-CNN approach, which recognizes the gestures effectively.

3.1. Data Acquisition

Lightweight IMU sensors, capable of measuring accelerated and angular velocity, are selected for precise motion tracking. IMU sensors have Bluetooth communication options, allowing them to transfer information to a data collection device. An IMU sensor should be attached to each palm and every finger. It is necessary to ensure that the sensors are firmly fastened in order to reduce movement artifacts and guarantee a snug fit for the signer. Then, researchers should ensure that the sensors are positioned and aligned correctly on each finger and palm. Finally, it is necessary to calibrate the IMU sensors for the correction of any discrepancies or aberrations in their measurements.

Instead of vision-based systems, this research focused on using gloves to measure hand and finger positions or gestures. Sign language users may find this approach less natural, but it gives precision, control, adaptability to varied conditions, and accessibility. Despite potential downsides, these attributes certainly influenced our decision to use gloves for data collection.

The data are obtained from the glove sensors for use in capturing gestures from hand variations. Figure 2a and specifications detail the glove-based wearable devices used to identify varying hand gestures, like hand movements and orientations. It is necessary to gather information as soon as the wearer makes the desired finger and hand movements.

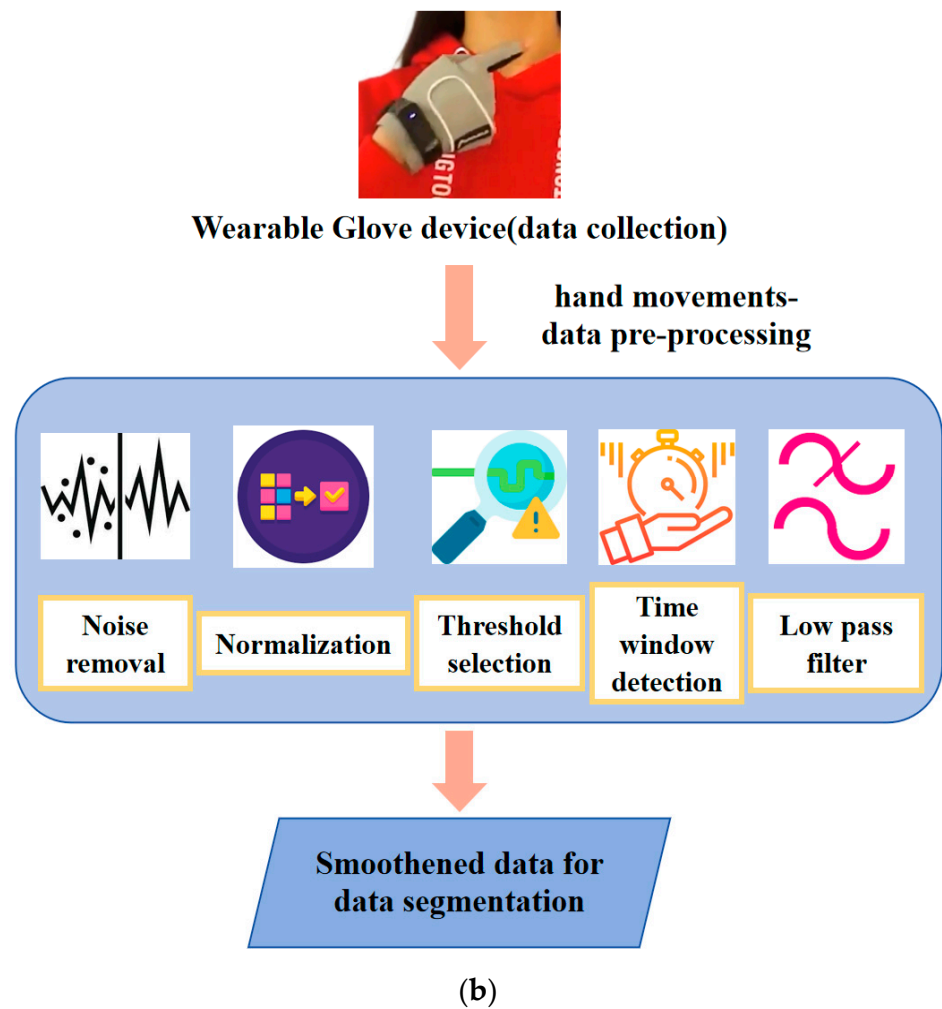
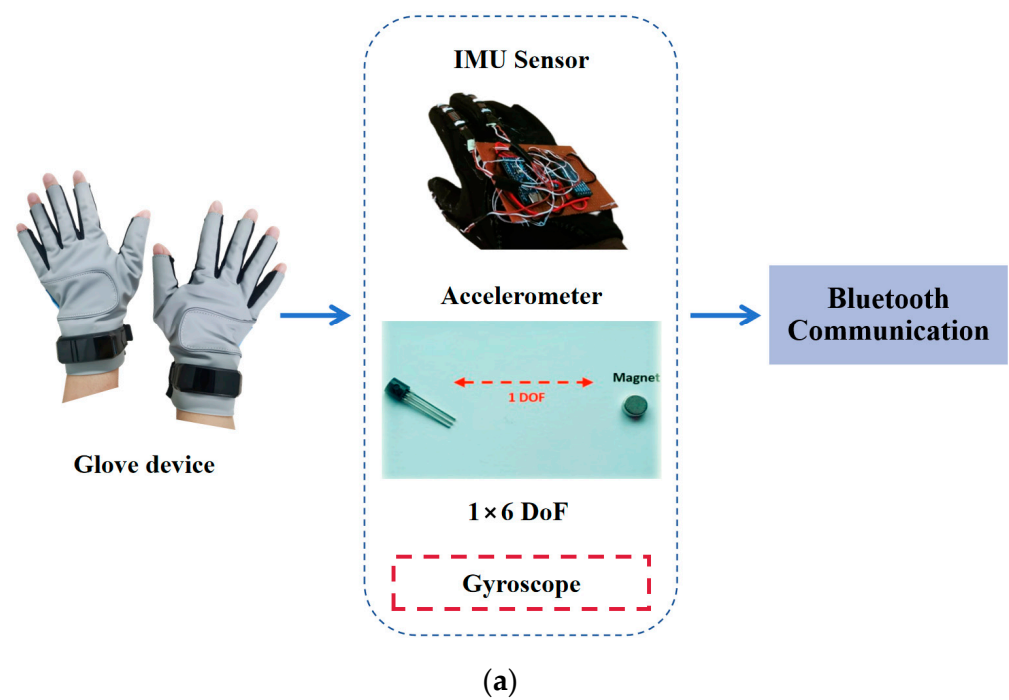


Figure 2. Cont.

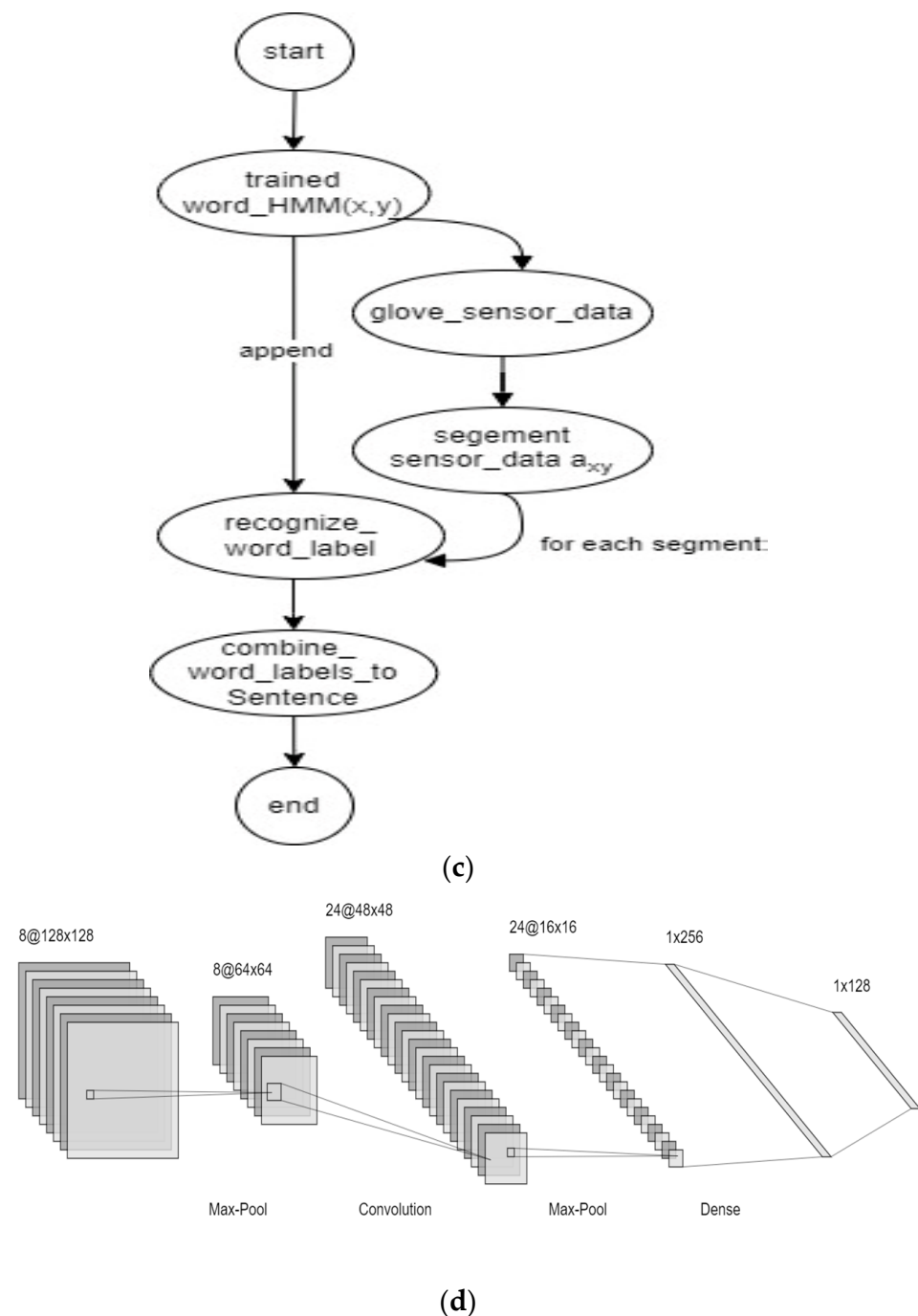


Figure 2. (a) Design of a wearable glove system. (b) Schematic representation of data preprocessing. (c) Control graph representation of the pseudocode recognizing word sign label using HMM. (d) Structure of convolution networks.

The ethical considerations related to the use of wearable devices for sign language recognition emphasize user autonomy. Along these lines, it is necessary to provide clear information about data usage policies and allow participants to make informed decisions about their involvement. These measures aim to uphold ethical standards, safeguard user privacy, and respect the autonomy of individuals involved in the study.

3.2. Data Preprocessing

A wearable glove device collects data by using technology with specified hardware details when each finger is flexed while making various sign language gestures. The

operation consists of storing large quantities of information, which is viewed as noise produced by the fingers shifting positions during each gestural movement. Before being input into the network, the measurements obtained from the gloves are preprocessed and converted into the features specified in this study. The features mentioned above were meticulously chosen to capture pertinent data regarding hand movements and gestures; the network subsequently employs this information for recognition.

Figure 2b illustrates the preprocessing techniques used for recognizing hand movements via wearable glove devices. It is necessary to reprocess the data collected from sensors in order to enhance its quality before segmentation. Common preprocessing procedures include noise removal and normalization to provide consistent and clean input. Optimal threshold levels that can assist in differentiating between hand gestures and prolonged resting conditions are selected. The threshold may differ depending on the particular sensors employed and may depend on sensor readings, including acceleration or location. By applying the chosen criteria to the sensor data, time windows are determined wherein hand movement is likely to occur. Various factors, such as external interference or inaccurate sensor readings, can cause noise in sensor data. By reducing noise and smoothing out the data, the low-pass filtering technique makes spotting important gestures simpler.

Using a low-pass filter in the preprocessing stage helps to remove high-frequency noise components, such as sensor noise and signals. This can improve the quality of input data and enhance the system's robustness to noise.

The PLD-CNN model utilizes a technique known as progression learning to enable the system to learn and adjust to the latest information over long periods of time. By changing its representations based on evolving data and variances in hand gestures, this incremental procedure for learning helps the model become more robust and accomplished by updating its representations.

Low-pass filtering analyzes every temporal and spatial data point when used on sensor data. This method considers signal intensity values within a specific time window or region around each data point. The filter computes a weighted average of all values in the window, giving those nearer to the filtered data point greater weight. The data point's new value in the filtered signal is the weighted average. The low-pass filter attenuates high-frequency noise, representing rapid alterations in the data. By averaging the noise over time, the filter reduces the high-frequency fluctuations in the signal. The filter maintains low-frequency components, representing the underpinning patterns or slower changes in the signal. The beginning of a possible gesture is indicated by the sensor data crossing the threshold, and the end of the potential gesture is designated by the sensor data falling below the same threshold. A low-pass filter smooths the signal, slows rapid changes, and retains the lower-frequency components while removing noise at higher frequencies from sensor data.

3.3. Data Segmentation

The information is divided into discrete gestures. These are divided into subwords, representing the basic units of sentences in sign language. During the division of words from a continuous sentence, the performance of the signer in maintaining the contraction of their muscles according to the sentences used for communication is assessed. The IMU palm sensor, with its 1×6 DoF specifications, captures finger variations, including thumb, index, middle, ring, and pinky (little finger) movements, along with finger stretching, hand movements, and overall hand posture during the learning phase of gesture translation between continuous subwords. Each subword fragment is subjected to feature extraction in order to produce sets of features representing continuous sign components, which are then used as input for the appropriate PLD-CNN algorithm at the element level. The next phase of subword-level identification integrates these elements after each has been analyzed individually. For additional processing, the segmented motions might be used as input for recognition models such as deep convolutional neural networks after performing feature extraction. Every sentence consists of numerous words, and each signal is signified

accordingly. The goal of this step is to segment each indication individually. There will be n corresponding segments in each sentence due to segmentation.

High-dimensional models (HMMs) are frequently used in sequential data modeling and can be used to identify patterns in time-series data. This system most likely uses HMMs to simulate the sequential nature of sentences that have been communicated through sign language. In HMMs, every movement or gesture can be categorized as a state, and transitioning between states is equivalent to transitions between gestures. HMMs can assist in segmenting the input data into meaningful units, such as individual signs or words, and in finding the sequence of these units that is most likely to produce a complete sentence.

A collection of states in the hidden Markov model for a segmentation task that represents various data segments or patterns for a gesture sign language with one gesture x to another gesture state as y is represented in HMM as a_{xy} . Each condition may be associated with a specific type of region. The features of the gesture within each segment are represented by the observations connected to each state. The emission probability $b_i(k)$ represents the probability of emitting an observation of finger variation k when in state x . The forward method calculates the likelihood of witnessing a succession of gestures, given the models used. It employs recursive equations to obtain forward probabilities $\alpha_t(i)$. Using these observations, the Viterbi algorithm determines the most probable series of hidden states, with the recognition of subwords set to $\alpha_t(i)$ and $\Psi_t(i)$. These probabilities determine how likely it is for one segment to transfer to another, that is, for a signer to transition from one gesture to another for identifying the continuity of words to make a sentence. Hidden Markov models (HMMs) represent individual phrases or gestures, collecting statistically significant trends and associated transitions. By predicting the probabilities of word sequences, these models, when paired with linguistic models, assist in the recognition of cohesive and grammatically accurate sentences in sign language. To efficiently model permissible sequences of words (sentences) in sign language, this research uses a combination of linguistic models and HMMs for individual gestures.

Pseudocode: Recognizing word sign label using HMM

```
Word_HMM = load _trained word_HMM( $x, y$ )
sensor_data = capture_glove_sensor_data()#input
segments = segment_glove_sensor_data(sensor_data for  $a_{xy}$ )#segment
recognized_labels = [ $\alpha_t(i)$ ,  $\Psi_t(i)$ ]
for each segment in segments:
    recognized_label = recognize_word_label  $\times$  (segment, word_HMM)
    recognized_label.append(previous recognized_label)
sentence_label = combine_wordlabels_toSentence(recognized_wordlabels)
recognized_Sentencelabel()
```

This pseudocode illustrates the key phases utilized in the process of using glove-sensor data and HMM to recognize sign movements and create a sentence sign label.

The explanation of pseudocode represents the recognition of word sign labels by using HMM with glove sensors to record each phrase as a sequence of sign language gestures. Learning patterns and characteristics extracted from input gesture sequences are used to train the PLD-CNN model in order to recognize these words. Using extracted features to identify word boundaries in the input sentence, the model generates recognized words. Subword extraction breaks down each word gesture towards subwords, which may be sign language letters or phonetics. The PLD-CNN model processes each word-containing gesture to extract subword-related unit information. The model outputs recognize subwords in order to capture gestural characteristics, including finger movements, hand forms, and trajectories. The model recognizes each subword unit to obtain a finer gestural word analysis and a better understanding of the area overall.

In sign language recognition, the applied control graph representation in Figure 2c provides an overview of the procedure used to recognize word sign labels by utilizing HMMs. The steps in this process involve loading a pre-trained HMM model, gathering sensor data from a glove device, segmenting the data, and detecting word labels for each segment

using the HMM model. After that, the detected labels are joined to generate a sentence label, which makes it easier to understand the meaning of sign language utterances. This algorithmic approach uses convolutional neural networks (HMMs) to describe temporal relationships in gestures and improve the accuracy of sign language recognition systems.

Once all time steps have been completed, it is necessary to revisit the $\alpha_t(i)$ and $\Psi_t(i)$ values to identify the most likely order of concealed states. The path that maximizes the combined likelihood of the observed gestural information and the state sequence corresponds to this. The word sign label is the classification given to a particular sign or gesture that denotes a word in sign language. These labels make the recognition and classification of particular indications within a sentence easier. The transition probabilities used in segmentation are often created to reflect the possibility of switching between segments. The possibility of observing particular finger points and characteristics in a given state is defined by probabilities. These estimations capture the features of each segment via segmentation tasks. Continuous glove-sensor data and gestural information are divided into subword indication portions. The next step is to label the segmented information accordingly, and the word sign labels are recognized via sign language recognition using glove sensors and the HMM segmentation technique. Continuous sensor input is segmented into different sign gestures, and each segment is then associated with a particular word sign label depending on the trained HMMs. The system can identify and categorize signs in the glove device data because the HMMs represent the variations at the time of each sign. The annotators should be fluent in sign language to appropriately label every sign in context. It is also necessary to train a unique HMM for the sign labeling of each subword using the annotated data. The HMM simulates the time-varying patterns of each sign based on the retrieved features.

The sign language recognition method presented in this research, in which sequence modeling often uses probabilistic HMMs, relies on HMMs. These models can describe gesture-related temporal dynamics in sign language recognition in order to capture sign language's sequential character. Typical HMM model preparation and training software includes a Python module called "hmmlearn", which implements HMMs.

The PLD-CNN model used in this study recovers words and subwords from glove-sensor sign language motions. For example, a sequence of sign language gestures captured using glove sensors represents the sentence: "I have a headache and cough."

Word: "I", "have", "a", "headache", "and", "cough".

Subwords: {"I"}, {"h", "a", "v", "e"}, {"a"}, {"h", "e", "a", "d", "a", "c", "h", "e"}, {"a", "n", "d"}, {"c", "o", "u", "g", "h"}.

Deep convolutional neural networks evaluate movements in order to identify word boundaries and break them down into letters or phonetic components. Deaf and hard-of-hearing people can better communicate by recognizing and interpreting sign language sentences.

HMMs are commonly utilized in sign language recognition systems to capture the temporal correlations and sequential patterns inherent in gestures. The recognition process can account for sign language temporal elements like gesture order and transitions using HMMs, which complement deep neural networks.

3.4. Feature Extraction

One of the primary goals of utilizing conventional machine learning algorithms is feature engineering. As the inputs for the specific machine learning model features, the standard statistical features utilized in the study of data from wearable glove-sensor devices for sign language recognition, including gestural information collected by peripheral glove sensors for the recognition of sign languages, encompass Mean Absolute Value (MAV), variance, and standard deviation. These features are frequently utilized as input features for training ML algorithms or for additional analysis since they might reveal important details about the nature of the sensor data.

Deep convolutional neural networks can extract hand gesture features from sign language inputs. Let us denote an input image depicting X , a hand shape made during a sign language gesture, as X . This image is typically represented as a matrix of pixel values, where each pixel encodes the intensity or color information at a specific spatial location. Deep CNNs start with convolutional layers that learn to recognize and extract low-level characteristics like angles, boundaries, and texture patterns from the input gesture images. Convolutional operations build feature maps by sliding a tiny filter, called the kernel or feature detector, through the input image and by performing element-wise multiplications and summing. This operation is mathematically expressed in Equation (1) and Equation (2):

$$C(i, j) = \sum_m \sum_n X(i + m, j + n) \cdot F(m, n) \quad (1)$$

where $C(i, j)$ is the value of the feature map at the position (i, j) , $X(i + m, j + n)$ represents the pixel value of the input image at the position $(i + m, j + n)$, and $F(m, n)$ denotes the value of the filter at the position (m, n) . Deep CNNs extract discriminative features that capture the visual characteristics that differentiate one sign or gesture from another.

$$M_{k,t} = \left\{ \left(x_{k,t}^1, y_{k,t}^1 \right), \left(x_{k,t}^2, y_{k,t}^2 \right), \dots, \left(x_{k,t}^n, y_{k,t}^n \right) \right\} \quad (2)$$

This allows the network to distinguish hand shapes and movements, enabling sign language gesture recognition. The same applies to the motion trajectory of a particular keypoint k in frame t , where n is the total number of points in the trajectory. Deep CNNs' output features are abstract representations of input motion patterns and can be used for gesture detection.

Mean Absolute Value (MAV):

In a segment of sensor data, the MAV determines the overall relative magnitude of each data point obtained from the fingers and each palm movement obtained from the gloves. This method is a measurement of the average intensity or frequency of the signal. MAV can assist in capturing the total power or force of hand motions during a gesture using Equation (3):

$$MAV = \frac{1}{n} \sum_{i=1}^n |G_i| \quad (3)$$

where n represents the total number of gestures in the segment and G_i represents the individual data points of identified gesture samples.

Variance (Vr):

Variance measures how evenly or unevenly data points are distributed within a segment. This parameter gauges how widely apart data points are from the mean and sheds light on how widely spaced-out sensor measurements might vary. It can assist in differentiating between motions that vary in their degree of reliability or predictability.

$$Vr = \frac{1}{n} \sum_{i=1}^n (G_i - \mu)^2 \quad (4)$$

μ represents the mean of the gesture sample data in Equation (4).

Standard deviation:

The square root of Vr yields the standard deviation, and the average departure of points of data from the mean is measured. The standard deviation indicates the range of the data and is closely associated with variance.

$$sd = \sqrt{Vr} \quad (5)$$

Equation (5) demonstrates that sign language, which utilizes the properties of finger motions, can be learned from data produced using glove sensors, which are utilized as feature inputs for ML-based predictive models.

3.5. Progression Learning for Enhancing Recognition Accuracy over Time

Sentences in sign language are first processed by being broken down into words. This progression method of learning continuously improves the system by producing a training set. As the system progresses through the phases, progression learning takes place, with each level expanding on the knowledge acquired in the one before it. The learning makes it possible to accurately recognize the entire sentence word by word using information from the glove sensor.

The PLD-CNN model uses progression learning implements with minimum latency values of <5 ms. A low-energy-efficient communication protocol like Bluetooth minimizes data overhead and transmission latency to conserve computational resources for the wearable devices utilized, along with employing feature extraction, memetic optimization, and efficiency analysis to improve computing efficiency. These solutions maximize system efficacy by balancing accuracy and computational efficiency to recognize sign language and manage the computational demands of wearable devices with GPUs.

The AI model better understands sentences in sign language because of the dataset in question. The progression learning network method accesses previously learned features for deep combination through the lateral connections of gestures identified, and it is appropriate for any progression learning environment, such as in the continuous observation of gesture signs during a sentence. The extracted dataset used for implementation purposes is divided into data for training (75%), and random testing data (25%). Incremental training drastically reduces the training time by allowing us to train the network solely for a new set of gestures obtained from the glove device and optimize the most recent fully linked layer without training the network again. The progression learning deep model, introduced here, is gradually trained on new inputs from wearable glove sensors using the suggested PLD-CNN technique, enabling the learning of recently obtained gestures without needing access to previous training information. Because progression learning enables the recognition system to gradually acquire context and increase accuracy as more motion is recorded, it can offer sentence-based classifications in sign language from wearable glove devices. Staged model training uses learning progressions with deep CNNs. With each stage, emphasis is placed on understanding gestures more precisely. The recognition algorithm improves in recognizing gestures at each level of progression learning. As stated, model training progressively utilizes PLD-CNN, emphasizing gesture understanding.

In order to record dynamic movements, sensors track joint angles, finger trajectories, and palm orientations at high sampling rates. Gloves use inertial measurement units (IMUs) and flexible sensors on each finger to capture tiny signing gestures and positions. IMU data track hand motions in 9-DOF, allowing the model to learn rotational motions and quick sign switches. Orientation heuristics enable the interpretation of hand directions as sign meanings based on the identified words in a gesture. These words are brought together to make a complete sentence for effective communication. A progression learning approach is ideal for sign language recognition because it can incrementally learn to distinguish between subunits and model longer sign sequences. Fluent transitions are learned through progression training. Large natural sign language video datasets with 100 sentences of sign language gestures are preprocessed to extract annotated gesture sequences. Fluent signing with authentic co-articulations and transitions serves a crucial role in training the progression learning models in real-world languages.

Hand movements are the only thing that the current glove-based technology takes into consideration; it does not consider facial expressions.

IMU sensors and wearables are given as inputs into Algorithm 1, where noise is eliminated, adjusted, and low-pass-filtered to improve data quality. Trained HMM word sign labels segment preprocessed inputs into gestures. Extracting key gestural properties allows for continuous sign–gesture representation. A progression learning system trains a deep CNN model using these attributes. The model learns linguistic and gestural information to improve recognition. Network parameters are optimized for accuracy using memetic opti-

mization. The recognition system is assessed using different criteria to ensure its efficacy. This iterative approach enhances the performance of sign language gesture recognition.

Algorithm 1: Progression Learning Procedure for Sign Language Gesture Recognition

Step 1: Input layer performs data acquisition

Collect input from IMU sensors and wearable devices.

Step 2: Data preprocessing

Preprocess the sensor data to enhance quality using noise removal, normalization, and low-pass filtering.

Divide phases into groups of 75% for training and 25% for randomly tested data.

Step 3: Data segmentation

Segment the preprocessed sensor data into discrete gestures.

Associate each segment of finger variations k with a specific word sign label based on the a_{xy} trained using HMMs.

Step 4: Feature extraction

Extract features from each gesture/subword in order to represent continuous sign gestures, varying based on hand variations a_x and a_y

Use dense convolutional layers to learn hierarchical representations of input gesture features.

Step 5: Progression learning

Train a model using the features extracted with the deep convolutional neural network learning procedure.

The model learns and adapts to the complexities of sign language gestures over time.

Step 6: Orientation learning towards gestural identification

Learn the network parameters and update them with information from new sentences S and gestures G_i , $i \in [1, 2, \dots, n]$

Word and subword recognition using $WER_{G(i)}$ from learned sequences enhances recognition accuracy and robustness.

Step 7: Memetic optimization algorithm

Finetune network parameters and improve accuracy in recognizing sign language gestures.

Step 8: Evaluation

Conduct performance validation using accuracy, precision, recall, and F1 score metrics, WER , and temporal consistency.

3.6. Orientation Learning for Tracking Glove Devices and Finger Signs

Orientation describes the direction that the glove device is facing, the fingers of the hand, or the IMU palm are facing. The goal of orientation learning is to precisely track how the glove device and the fingers on the hand are oriented. The investigation of orientation learning will allow for the proper detection of sign language motions and signs. A variety of hand and finger movements are used in sign language. Since the direction of the hand can vary for various signs, the glove device's orientation aids in identifying the precise gestures being made. It adjusts the degree of yaw caused by fluctuations in the finger by applying magnetic sensor data collected from the glove device. This adjustment ensures that the system recognizes the right term and interprets the movements correctly. Typically, the orientation can be estimated using the angle of an accelerometer, as part of the IMU glove device, and the weight components along its axes. A portion of subwords is used, with their orientation shifting over incremental time due to the wide range of progression sign gestures, and the initial and final orientations of a complete subword fragment are of the utmost significance in sentence-level communication related to sign meanings produced by the signer. Calculations that utilize the inertial measurements along finger and palm reading lines are commonly used to estimate orientation from accelerometer data. It is typical to estimate orientation by computing the roll, pitch, and yaw degrees. These angles describe the orientation of the device in three dimensions. Based on accelerometer readings, the following formulae can be used to determine the roll, pitch, and yaw angles using the equations below (6 to 8):

$$\Phi = \arctan2(a_x, a_y) \quad (6)$$

where a_x represents the finger movements along the x -axis and a_y represents the finger movements along the y -axis. This is followed by the pitch angle analysis of the rotation of glove devices to identify gestures around the x -axis. The yaw degree represents the finger rotations around the z -axis.

$$\theta = \arctan2\left(-a_x, \sqrt{a_x^2 + a_y^2}\right) \quad (7)$$

$$\omega = \arctan2(b_x \cdot \sin(\Phi) - b_y \cdot \cos(\Phi), b_x \cdot \cos(\Phi) + b_y \cdot \sin(\Phi)) \quad (8)$$

where b_x represents the magnetic sensor readings of a wearable glove used to correct the yaw degree related to all five finger variations, ranging from a to e, in order to identify the correct word through learning progression.

Individual gestural identification:

The recognition approach is initially taught to recognize particular sign language motions. For instance, the paradigm in column 1 identifies these movements individually but ignores the context in which they occur. Progression networks that inhibit catastrophic recollection by developing an entirely new neural network, which acts as a segment associated with every gesture being identified and enables transmission through the creation of extensions to the characteristics of sections learned earlier, incorporate these requirements. Let G be the set of sign language gestures, where each gesture is represented as G_i , $i \in [1, 2, \dots, n]$ and where n represents the number of sentence gestures. Let $R(G_i)$ represent the recognition result for the gesture G_i that can be obtained from the previously trained deep CNNs. Then, the structure of CNN is illustrated in Figure 2d.

The recognition of a sentence S consisting of n gestures can be represented, as given in Equation (9):

$$S = [G_1, G_2, \dots, G_n] \quad (9)$$

Deep CNN parameters must be initialized, and a recognition method must be used to determine the next task based on sensor duties, which are focused on finger and palm gestures, for use in continuous learning. We must develop a mechanism with which to effectively increase the recognition model's capacity to adapt when learning new tasks. Likewise, there is a pressing need for a technique to collect, maintain, and use knowledge in order to learn subsequent tasks without significantly degrading the previously learned information. Similarly, the identified result for sentence S is given in Equation (10):

$$R(S) = R(G_1) \oplus R(G_2) \dots \oplus R(G_n) \quad (10)$$

where $R(G_i)$ reflects how well each gesture in the statement is recognized. \oplus represents the operation that concatenates or combines the recognition findings of different gestures to create the classification outcome for the complete sentence. The network starts with a single column, representing deep CNNs with multiple layers l , and a hidden activation $h_i(1) \in R(n_i)$, where n_i represents the number of gesture units identified at layer $i \leq l$ and the corresponding trained parameter setting for translation convergence. When moving to the second layer of gesture recognition related to finger orientation, the previously learned gestures for words are frozen $\varphi(G_1)$ for further learning, and the new column with the next set of gestures selected for the second phase of sentence communication is randomly instantiated, and the current layer learns inputs from $h_{i-1}(1)$, where $h_{i-1}(2)$ represents the previously identified subwords. The model is enabled by progression learning to excel at identifying signs within words while considering context and communication flow. As previously learned finger traits are used in several activities, learning in order to transfer becomes more effective. This method ensures that prior task knowledge is preserved and is impervious to catastrophic forgetting.

Subword recognition:

The model concentrates on identifying subwords from whole sentences or transitioning between motions in the second stage, which aids in capturing more intricate linkages between movements. The recognition system generates a series of recognized words or signals based on the input glove–sensor data. The word error rate (*WER*) is useful when evaluating the recognized sequence’s general efficacy while also accounting for insertions, deletions, and substitutions. As it considers the location and kind of inaccuracies committed in the recognized sequences compared to the reference sequence, it offers a more thorough evaluation than the use of correctness alone:

$$WER_{G(i)} = \frac{Ins + Del + Sub}{T} \times 100 \quad (11)$$

where *Ins* represents the number of insertions of accurate words into the sequence, *Del* is deletions like missing words or signs in the recognized sequence, and *Sub* is the substitution of mismatched or identified words. *T* refers to the total number of signs in the references. The recognition performance increases as the *WER* decreases (refer to Equation (11)). A larger *WER* implies greater recognition of errors, while a *WER* of 0% indicates flawless recognition (no errors). *WER* is useful when measuring the recognized sequence’s overall quality while also accounting for insertions, deletions, and substitutions. As it considers both the location and type of inaccuracies committed in the recognized sequence compared to the source sequence, it offers a more thorough evaluation than correctness alone.

Sentence analysis:

This method understands that gesture combinations can indicate several words and that order matters by aggregating the previously analyzed phases. Then, more phrases are subsequently added to the process, with each building on what was learned at the prior level. These techniques should become more adept at identifying minute features in sign language gestures.

This stage analyzes the sign language sentence’s entire context order by aggregating subwords. Considering the setting and similarities discovered at each level of progression learning, the ultimate recognized result is a logical interpretation of the communication in a given sentence. With progression learning, the recognition system can gradually increase its comprehension of sentences in sign language, enhancing its accuracy and contextual awareness as it advances through stages. It is necessary to utilize validation data and suitable measures, including accuracy, precision, recall, and the F1 score, to assess each model’s performance. These metrics guarantee that recognition accuracy improves at each level.

3.7. Memetic Optimization Algorithm for Improving the Precision of the Recognition System

The parameters of the recognition network model can be adjusted, and its accuracy can be increased using memetic optimization methods. Memetic optimization improves the detection of sign language motions, improving success in correctly identifying and converting signs into text by iteratively improving the model’s performance using local search approaches.

The memetic optimization approach improves sign language recognition systems by optimizing neural network model parameters. This optimization method iteratively improves model performance using evolutionary algorithms and local searches. The method adjusts network parameters based on training and validation data to enhance recognition accuracy and reduce sign language translation problems.

Impact on precision enhancement:

Setting up parameters:

The memetic optimization algorithm adjusts neural network parameters to better capture the complicated patterns and nuances of sign language movements, improving recognition accuracy.

Iterative refinement:

The algorithm updates and optimizes the model's configuration, utilizing fresh data and user input to improve performance.

Local search approaches:

The algorithm refines the model's ability to make decisions by using local search approaches, resulting in more accurate sign language gesture and sentence recognition.

Reducing translation difficulties:

The optimization process improves the system's ability to identify and interpret sign language gestures, facilitating the communication abilities of hearing-impaired people.

In this optimization phase of training, the objective functions need to be designed to boost training and validation accuracy and improve the translational computation cost of the proposed progression learning.

In progression learning for sign language recognition, the optimization problem induces the iterative enhancement of a neural network model's performance by adjusting its parameters via memetic optimization techniques. The primary objective of doing so is to optimize the system's precision in identifying and transcribing sign language gestures, facilitating more effective communication among users who are hard of hearing.

As an example, let us consider a sign language recognition system that encounters initial difficulties in accurately interpreting intricate gestures due to a lack of training information. By employing progression learning and memetic optimization techniques, the system progressively enhances the performance of its neural network model by adjusting its parameters in response to feedback received from validation data. As the system advances through distinct phases, it acquires greater proficiency in discerning hidden characteristics within sign language gestures, culminating in an upward trend in accuracy. The system learns from fresh data and improves the accuracy of its sign language identification performance at each level. The model is updated, and new information is added to improve sign language phrase interpretation and translation, improving communication for deaf and hard-of-hearing people.

The memetic optimization technique is employed to train the proposed neural network model using the parameter settings outlined below.

Initialization and fitness function:

The challenge in understanding hand motions in sign language recognition is turning them into meaningful signals or words. A sample of potential solutions—recognition models or algorithms—is developed to solve this issue. For applications that recognize sign language, recognition accuracy must be maximized. The technique operates well in comprehending and interpreting sign motions detected by glove sensors, as evidenced by a greater level of recognition accuracy $R(G_i)$. The solution space may be thoroughly explored thanks to the genetic algorithm element of the memetic optimization technique. It considers a wide range of options and investigates various strategies and setups for sign language recognition. The equation $fit(i) = recognition\ acc(i)$ shows the number of correctly recognized gestures (i) in relation to the total number of gestures.

Selection of local search methods:

Sign language recognition may need to be fine-tuned in order to capture the subtleties of hand gestures and movements effectively. The memetic algorithm incorporates local search approaches in order to enhance population-level individual answers for the words identified as having the best accuracy. These methods are intended to improve the recognition models' performance when dealing with certain gestures and variations. The fitness rating measures how successfully a proposed PLD-CNN recognition system or algorithm interprets hand gestures and converts them into words and meaningful sentences. More physically active individuals have a better probability of being chosen as parents, and it is necessary to create new candidate solutions using genetic operators such as crossover (recombination) and mutation. Genetic information extracted from two source solutions is combined during crossover to produce offspring. Individuals undergo minor random alterations as a result of mutation.

Iterative refinement: The memetic optimization approach iteratively improves the population's recognition models. It determines the most effective settings for various sign language movements, increasing recognition accuracy. Achieving a desired fitness level, running for a predetermined amount of time, or attaining a gesture G_i generations are instances of common termination criteria.

High-quality solutions: The algorithm converges to high-quality recognition models through global discovery and local exploitation. The sign language system used for recognition is improved and becomes more dependable thanks to these models, which are skilled at accurately distinguishing and interpreting sign language motions. If necessary, researchers should repeat the memetic optimization procedure repeatedly to significantly enhance the system's performance in light of fresh information or shifting demands.

In summary, the proposed methodology met the objective of reducing the translation challenges faced while performing sentence-based sign language communication among disabled people, achieving a very robust performance. Additionally, the PLD-CNN demonstrated that the progression technique is robust to all types of gestures, which are taken as learning features, in conflicting operations during sign language recognition with intelligent glove devices. Further, this model can efficiently perform translation for appropriate source and target regions. The accuracy of sign language recognition is improved over time via progression learning. This entails breaking down phrases into words, assembling a training dataset using a CNN, and incrementally enhancing the system's knowledge through learning over time. The method uses subword recognition, orientation tracking, incremental training, and staged model training to improve sign language recognition accuracy gradually. Memetic optimization eventually produces high-quality recognition models, which lessen translation difficulties and boost sign language communication precision, particularly for sentence-based communication for deaf and hard-of-hearing people.

Through the use of a progression learning approach, the model's capacity can be effectively increased by progressively adding more learning modules as needed. This enables upscaling to manage large and varied datasets. Identifying the best architectures and hyperparameters to fit expanding datasets is made easier by the supported memetic optimization. For use in the real world, the glove-sensor devices offer a reliable and useful way to record sign language inputs. The adaptable progression learning method allows for the sustainable addition of more recent linguistic data units by various components.

4. Results and Discussion

4.1. Data Source Information

The proposed system is tested on a very challenging dataset with extremely high levels of endurance. The dataset includes a demonstration of 100 sentences observed through wearable glove devices. The numbers of words and subwords are identified through the proposed PLD-CNN model, which includes individual words and sentences. The gestures presented by the signer vary. Indeed, the time needed for the training and recognition of each gesture is mentioned in the sources as being <5 ms. If the overall time window consists of 479 data frames, then it equals $479 \times 5 \text{ ms} = 2395 \text{ ms}$, meaning that the sign language gestures last for approximately 2.4 s. It is possible to create a wearable glove device for a sign language recognition system. Its descriptions are detailed in Table 2 with the sample sentence-based sign language recognition system model. Motion sensors like gyroscopes, accelerometers, and magnetic sensors are present in many glove wearables. The data gathered with smart gloves' wearable sensors contain information like finger stretching and flexing, hand movements, and posture, all of which are included in the data file's CSV form.

The dataset used for this research consists of sensor data collected from wearable gloves equipped with IMU sensors according to the specifications described in Table 2. The data source size contains 100 sentences for sign language recognition among hearing-impaired groups, with gestures selected from International Sign Language. Sign language motions with different hand movements and orientations may be included in the dataset,

which may also include common and uncommon sign language motions. The dataset's diversity exposes the model to different motions, improving its generalization to other sign languages and users. Demographics, sign language culture, and gesture performance may cause biases.

Table 2. Details of the parameters used in the proposed model.

Source	Description
Finger variations	a: thumb, b: index, c: middle, d: ring, and e: pinky
Palm IMU sensor	$1 \times 6n$ (degree of freedom)
Latency	$\text{sig} < 5 \text{ ms}$
Degree of freedom (9 dimensions)	5 finger mid-joint bending (0–100%) + 4 quaternion data (0.000–1.000)
Sensor sampling rate	Finger: 200 Hz and IMU: 1000 Hz
Battery duration with charging device weight	>8 h; USB (about 90 min); 70 g total
Finger curvature sensor types	5×WULALA-CS1G

Labeling sign language gestures incorrectly can bias training data results derived from annotators' subjective assessments of gestures in specific sign language semantics, leading to data misclassification. Since the data-gathering technique mostly captures gestures used in sign language by fluent signers in controlled situations, the model may not be generalizable to different signers and environments.

The potential environmental impact of the proposed wearable glove devices, such as the flexible nylon/spandex materials, which are used in production and disposed of at the end of their lifecycle, has been assessed with sustainable design materials and found to be compliant with regulations.

There is a tag that identifies the data entering the column. Quantitative values might be represented by the numbers listed in the other columns, labeled as q0, q1, q2, and q3. The categorization information may be associated with a variety of fingers, including the thumb, index, middle, ring, and pinky. The database of the intelligent sign language recognition system contains the data obtained from the sensor, as shown in Figure 3a,b.

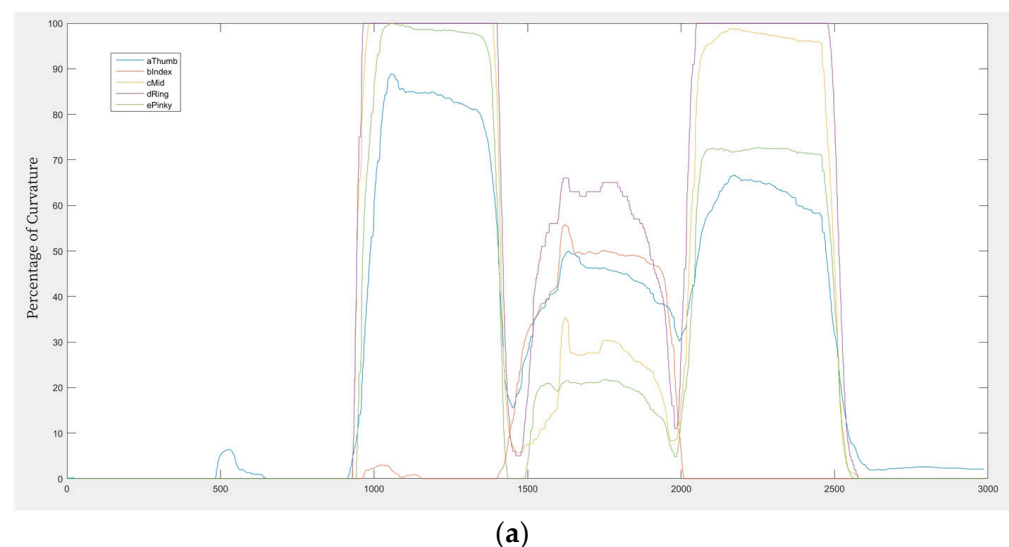


Figure 3. Cont.

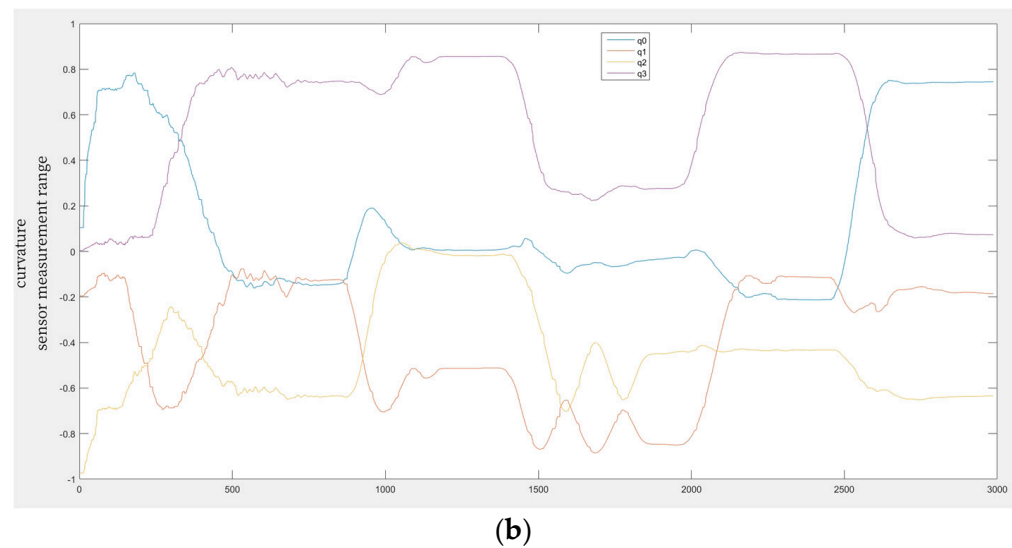


Figure 3. (a) Normalized data (in percentage of measurement range) for bending angles of 5 fingers in the sign sentence “I have a headache and cough”. (b) Quaternions for palm in the sign sentence “I have a headache and cough”.

Systems demanding significant computational capacity require more advanced processing devices and a longer processing time in order to recognize signs. Signers can now convey gestures through gadgets and processes because sensors can recognize hand motion. The finger curvature sensor measurement range is specified as -30° to 180° , with the sensor’s sampling rate for each finger taken as 200 Hz and that for IMU as 1000 Hz.

Table 2 details the glove-based wearable device specifications used to identify varying hand gestures, such as hand movements and orientations. In sign language recognition research, these designations may refer to the wearable glove device’s sensor-equipped fingers. The recognition system processes finger movement data extracted from finger sensors. Thus, the labels “a” to “e” assist in identifying the fingers that every set of sensor data relates to, where a is the thumb, b is the index finger, c is the middle finger, d is the ring finger, and e is the pinky finger. This enables the correct interpretation and evaluation of sign language hand gestures.

The experimental glove fitted flex sensors on each finger in order to measure bending angles from 0 to 90 degrees, which were used in addition to the specifications in Table 2. The glove also exhibited palm and wrist inertial measurement units (IMUs) with three-axis accelerometers and gyroscopes for six degrees of freedom. Bluetooth Low Energy (BLE) provided a solid connection between the glove and the computing device for up to 10 m, as shown in Figure 4a,b. Flexible nylon/spandex made the glove comfortable and breathable for long-term usage.

We used a Bluetooth USB receiver with Bluetooth version 4.0 and a 2 Mbps data transfer rate for data reception on a personal computer. Computing device specifications varied per platform. A 2.5 GHz Intel Core i5 processor (Intel, Santa Clara, California, United States), 8 GB of DDR4 RAM, 256 GB SSD, and NVIDIA GeForce GTX 1650 or GTX 1660 were sufficient for PC development. The storage device was an Android 10 smartphone with a Qualcomm Snapdragon 665 processor (Qualcomm, San Diego, California, United States), 4 GB of RAM, and 64 GB of storage. The software environment included Visual Studio 2019 Community Edition for PC and Android Studio 4.2 software for developing apps for Android, and this was implemented with MATLAB (R2021b). TensorFlow Lite was used for machine learning-based gestural identification. Tests were run using JUnit for unit testing and Matplotlib to visualize data to ensure experimental reliability and correctness. Due to these detailed parameters, the study’s sign language recognition outcomes are reproducible and comparable.

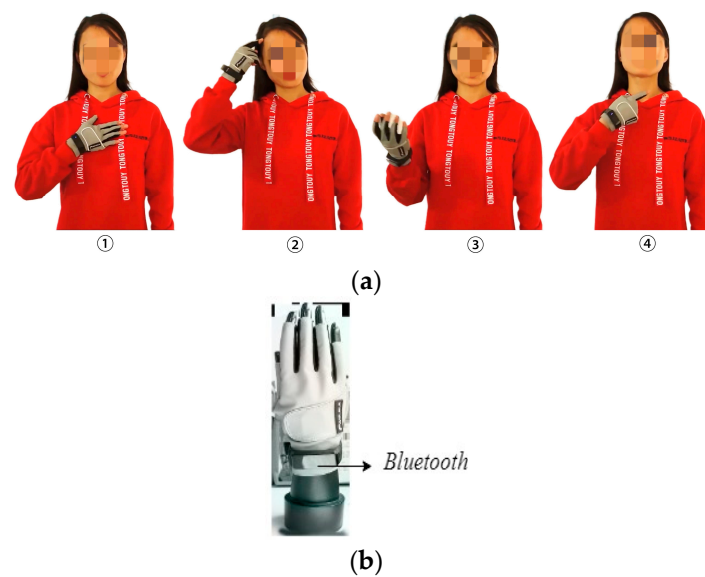


Figure 4. (a) Illustration of a graphical representation showing sentence formation: “I have a headache and cough” (①: I, ②: head, ③:ache ④: cough). (b) Sample intelligent wearable glove device (battery: >8 h with USB charging for 90 min; weight: 70 g).

The user interface considerations related to the design of wearable glove devices include varying gestures for translating signs or performing control tests for devices. There are ergonomic factors, such as a maximum battery weight of 70 g, to minimize hand fatigue. Finger and wrist mobility detection found good acceptance among users because it uses mid-joint bending variations to accommodate all user hand sizes.

4.2. Evaluation Metrics

The generated features are then sent into an upgraded learning environment for sentence-based communication recognition. The evaluation metrics considered for confirming the effectiveness of the proposed PLD-CNN system are accuracy, precision, recall, and F1 score. The results are assessed against the HDL-ASLP [19], pa-TGCN [20], and CNN-LSTM systems [23]. These techniques process sign language efficiently with little computational difficulty, but they face limitations related to translation accuracy. In order to provide a comprehensive understanding of system performance, a diverse set of evaluation metrics is used on two variants, such as the no. of training epochs and the no. of recognized gestures from wearable devices, to assess their accuracy, precision, recall, and F1 measure. In addition, to enhance contextual understanding of wearable devices, temporal consistency through PLD-CNN and WER is used as a performance metric that ensures freedom from potential biases. Finally, transparency in reporting the evaluation metrics and achieved results facilitates efforts to reach conclusions based on research findings.

4.2.1. Accuracy

A simple metric called accuracy counts how many occurrences, including actual positives and negatives, are properly identified in all instances. The formula for the recognition accuracy of sign language is given as Equation (12):

$$Acc(\%) = \text{Number of correct predictions} / \text{total number of predictions} \quad (12)$$

True positives (TPs) consider the number of positive instances accurately anticipated as being sign language movements. True negatives (TNs) represent the percentage of instances accurately anticipated as being negative, such as correctly identifying non-gestures or other classifications. False negatives (FNs) represent the number of situations that are mislabeled as positive, such as when gestures are mistakenly identified as positive when they are

not. False negatives (FNs) represent the number of situations misclassified as negative, indicating the failure to see movements when they occur. Similarly, false positives (FPs) represent instances that are misclassified as negative, indicating the failure to detect hand movements when they occur. The comparison of accuracy is shown in Figures 5 and 6.

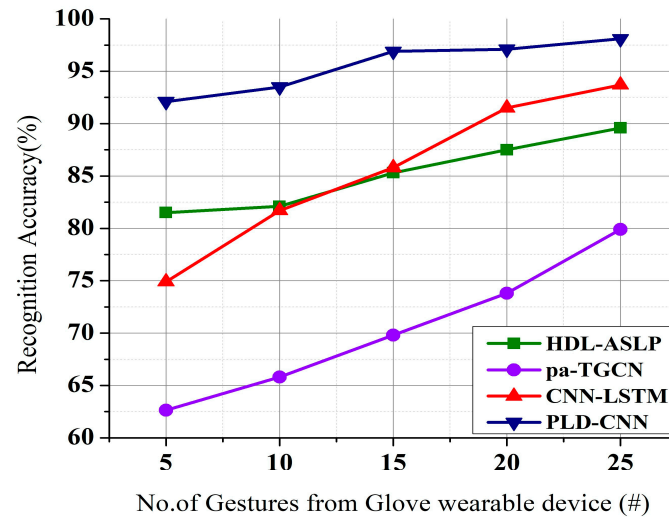


Figure 5. Accuracy based on recognized gestures from wearable glove devices.

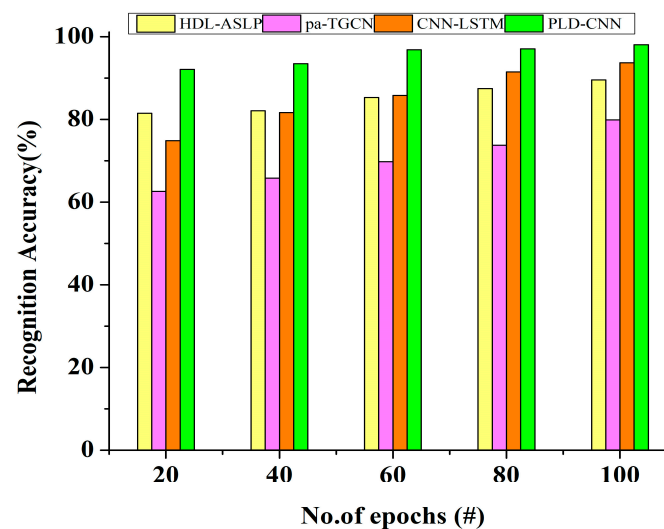


Figure 6. Accuracy vs. no. of epochs.

Allowing the system to learn and adapt over time and focusing on important information improves recognition accuracy without adding greater computational complexity. The PLD-CNN model uses memetic optimization to measure network performance and reduce issues with optimizing recognition. This optimization method balances recognition accuracy and computational complexity by fine-tuning system parameters and network performance to improve model accuracy without raising processing loads. The present study shows how design choices affect recognition accuracy and computational complexity.

4.2.2. Precision Analysis

The system's precision in predicting sign gestures is measured using Equation (13). High precision is necessary for the reliable recognition of sign language systems to reduce misunderstandings. This is because it means that when the ML algorithm anticipates a sign gesture, it is quite likely to be correct. Precision analysis measures how well the system

predicts the sign gestures people using the glove sensors will make. The comparison of precision is shown in Figures 7 and 8.

$$\text{Precision (\%)} = \frac{TP}{TP + FP} \quad (13)$$

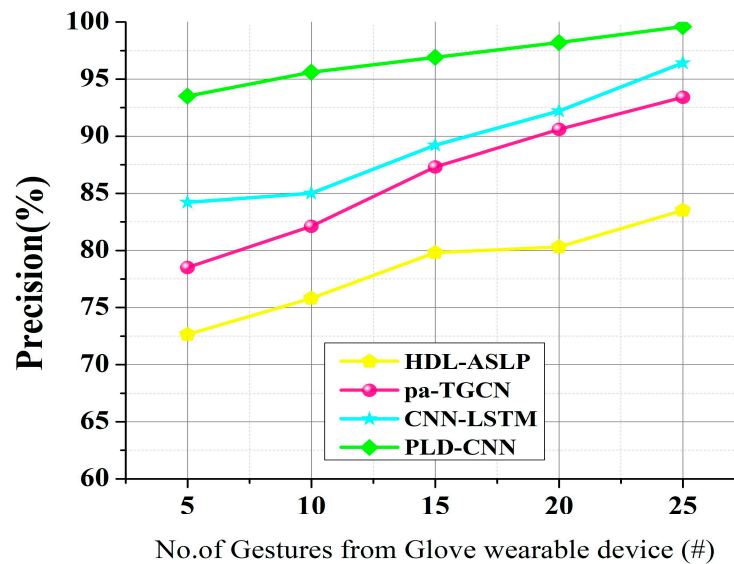


Figure 7. Precision based on recognized gestures from wearable glove devices.

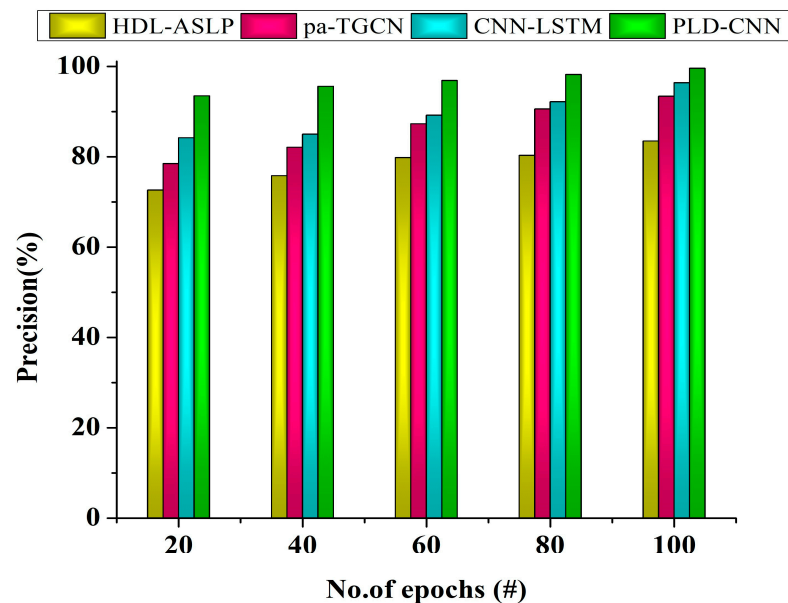


Figure 8. Precision vs. no. of epochs.

4.2.3. Recall Ratio Calculation

The recall ratio is a performance metric called the true positive rate (TPR). This metric is used to recognize sign language for sentence-based communication. All of these words refer to the same idea: the percentage of true positives indicates positive cases that are correctly recognized out of all genuine positive cases, and it is an indicator of the ability of a PLD-CNN system to recognize any pertinent gestural instance in a dataset correctly. Recall gauges the system's aptitude for accurately identifying sign motions made by the users of glove sensors, determining the system's sensitivity to movement. The comparison

of recall ratio is shown in Figures 9 and 10. Recall (sensitivity/TPR) is calculated using Equation (14), as follows:

$$Recall(TPR) = \frac{TP}{TP + FN} \quad (14)$$

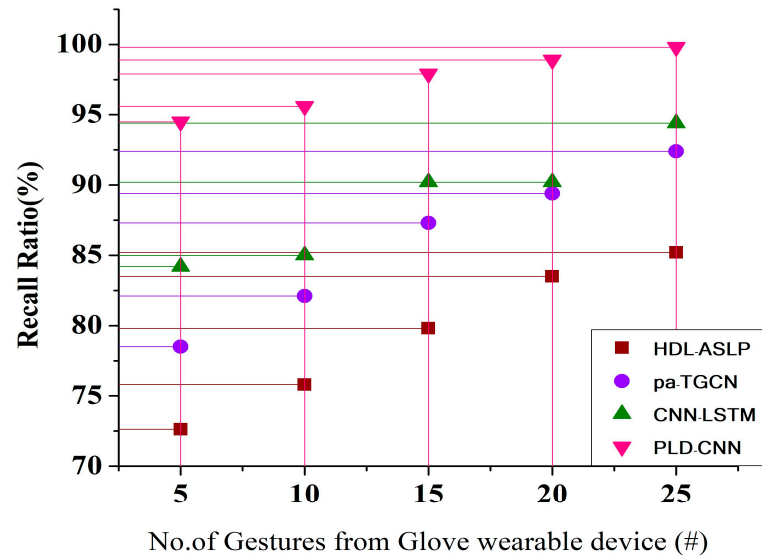


Figure 9. Recall vs. recognized gestures from wearable glove devices.

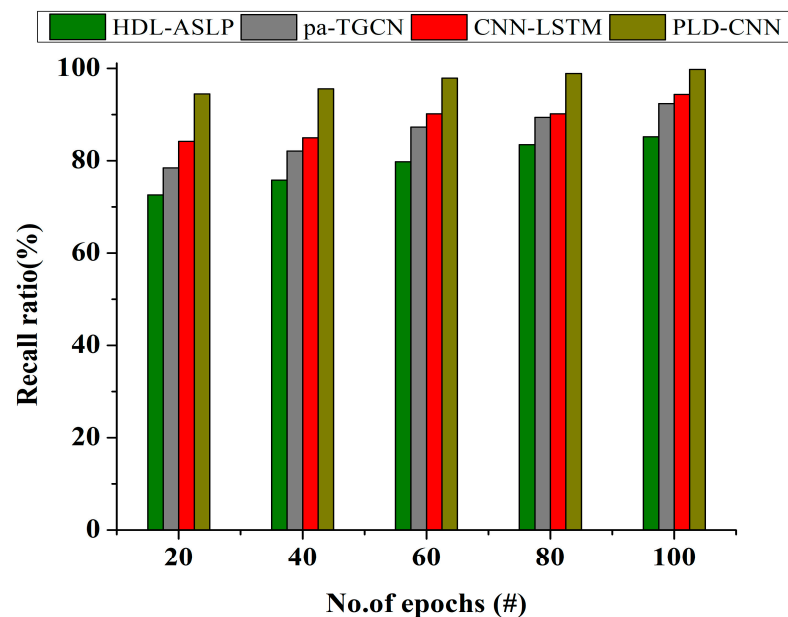


Figure 10. Recall vs. no. of epochs.

4.2.4. F1 Score Calculation

The percentage of accurate positive forecasts made among all positive predictions is known as precision. In sign language recognition, precision (P) refers to how accurately the system predicts certain sign motions. Out of all of the signs the system anticipates, P calculates the percentage of correctly recognized signs. When a system anticipates a sign, this is likely to be accurate if system precision is high. Recall is the percentage of predictions that come true from positive cases. Recall (R) is a measurement of a system's capacity to accurately identify and decipher all pertinent sign gestures made by the users of the glove sensors. This metric illustrates how sensitive the system is in terms of

identifying indicators. A high recall rate indicates that all pertinent signs are successfully captured by the system defined in Equation (15). The comparison of F1 score is shown in Figures 11 and 12.

$$F1 - Score = \frac{2PR}{P + R} \quad (15)$$

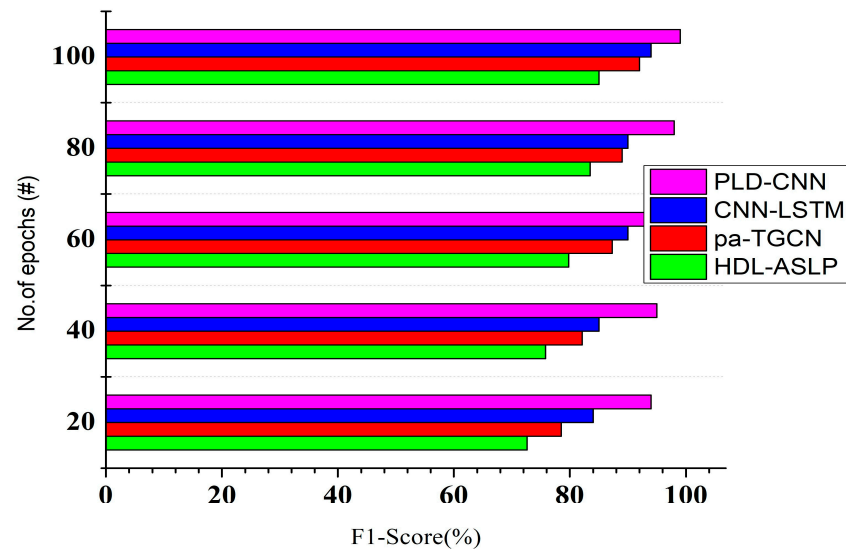


Figure 11. F1 score vs. no. of epochs.

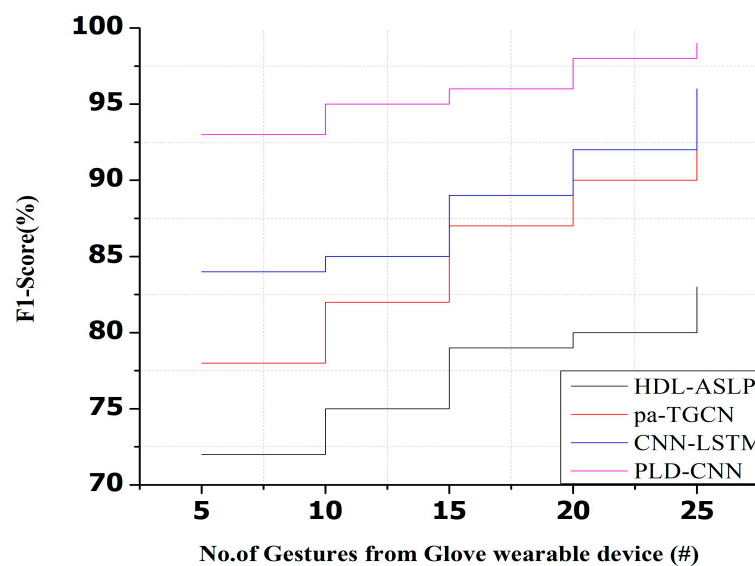


Figure 12. F1 score vs. recognized gestures from wearable glove devices.

Precision and recall are balanced into a single metric called the F1 score. It offers a mechanism to evaluate the system's overall performance while accounting for both FP and FN. A high F1 score implies that the system successfully balances precise recognition with thorough sign gesture coverage, achieving high precision and good recall. A high F1 score in sign language recognition indicates that the system is sensitive to detecting a variety of sign motions and accurate in recognizing signs, making it a useful indicator for determining the system's overall performance.

In real-world situations, the word error rate (WER) may be a better metric than accuracy, precision, recall, and the F1 score for use by sign language recognition systems. The purpose of sign language recognition is to translate motions into text accurately. The

WER measures translation errors, which helps sign language communicators achieve fluency and contextual knowledge. The WER considers word insertions, deletions, and replacements in order to better assess system-wide performance. Effective communication requires the technology to transfer sign language and gesture meaning accurately. It is vital to evaluate the fluency metrics, with strong system consistency in timing and the sequencing of sign language gestures guaranteeing seamless and natural communication. This is something that the research emphasizes as being essential for situations that occur in the real world. Figures 13 and 14 illustrate the importance of WER and temporal consistency analysis for contextually understanding sign language recognition.

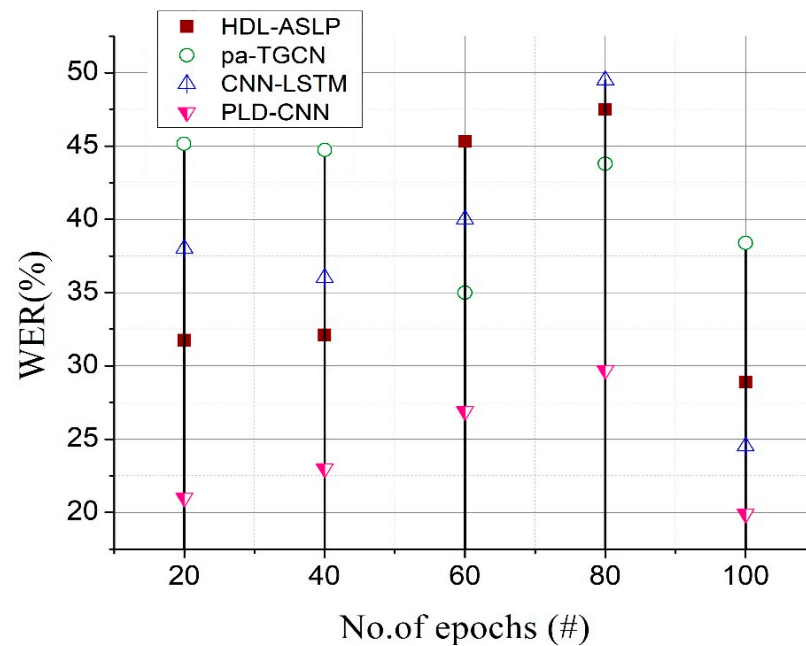


Figure 13. Analysis of WER in sign language recognition.

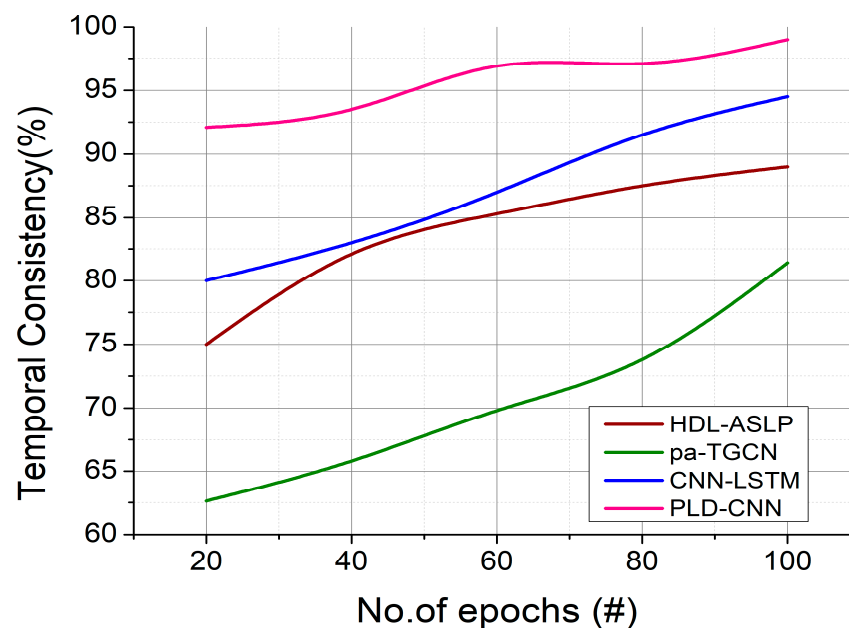


Figure 14. Analysis of temporal consistency in sign language recognition.

Temporal consistency and word error rate can help to resolve recognition system errors by revealing gesture sequence consistency and word or sentence accuracy. The model can

find patterns of gesture sequence inconsistency by evaluating temporal consistency, which may reveal instances in which the system struggles to comprehend gestures. Word error rate analysis can also identify commonly misconstrued or inaccurately recognized words or phrases, indicating that semantic problems or recognition algorithm improvements can increase system performance and accuracy.

The use of assistive technology, like connecting glove devices to desktops and mobiles, demonstrates the interdisciplinary nature of the research findings and wider considerations of user interaction and accessibility. Through communication between gloves and the PCs via the Bluetooth USB receiver, real-time data transfer is made possible, allowing analysis, processing, and comprehension via the utilization of PC platform applications. A Bluetooth USB receiver is not necessary when creating the glove device to link to an Android smartphone. This is due to the fact that most smartphones come equipped with built-in Bluetooth, meaning that an additional receiver is not necessary.

5. Conclusions and Future Work

The wearable glove device can transform sign language conversations into words and sentences through PLD-CNN analysis by recording hand motions and gestures using gloves and artificial intelligence. Additionally, neural models can be tuned to achieve better recognition performance through memetic optimization. The necessary preobservation of sensor devices is analyzed through IMU sensors. This is followed by the preprocessing steps, the segmentation procedure using HMM, and the feature extraction steps. These allow researchers to feed data into the PLD-CNN model for the assessment of gestures and to recognize the corresponding words. Subsequently, an appropriate sentence formation is produced through word label recognition via WER. The sign language recognition model achieves high levels of accuracy by fine-tuning its wearable glove-sensor parameters using the memetic optimization algorithm. This model also achieves fast convergence and mitigates translation complexities. The identified accuracy of the recognition model was 98.7%, with a precision of 99.6 and a recall (TPR) of 99.8%. Finally, there was an F1 score of 99% for sign language prediction accuracies regarding glove-identified gestures over a variety of epochs.

The possible cultural implications of employing technology to recognize sign language include the maintenance of linguistic variety or the strengthening of social biases. By providing tools for real-time translation and interpretation through videos, modern wearable devices, and mobile applications, technology can facilitate communication between deaf and hearing individuals, breaking down barriers and promoting social integration.

Regional sign language variations are needed for analysis, which is one of this method's main drawbacks. Diverse cultures and personal styles can affect sign language, which the system's training data may not represent. The lighting conditions of the employed wearable devices can also impair gesture recognition accuracy. These constraints show that the system must be refined and adapted to work in varied environments and groups of users in future work. There is a lack of research into the potential challenges related to wearable glove devices for sign language communication, i.e., lighting conditions and diverse participant populations, impacting the generalizability of our research findings.

In order to augment the system's robustness and inclusivity, subsequent research should concentrate on augmenting the dataset with regional sign language variations, integrating adaptive learning algorithms to enable personalized recognition; devising methodologies to accommodate diverse environmental conditions, and engaging individuals in the design process in order to incorporate user-centric enhancements.

The variations in lighting conditions while recognizing sign language should be the focus of future work.

Future research should also address privacy and security problems related to data handling. This entails using anonymization, imposing access limits, and encryption to protect sensitive sign language information gathered from wearable technology.

Author Contributions: Conceptualization, Y.L. and C.J.; methodology, Y.L. and C.J.; software, Y.L.; validation, Y.L. and C.J.; resources, P.C.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L. and C.J.; formal analysis, Y.L.; supervision, C.J.; project administration, P.C.; funding acquisition, P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Srinakharinwirot University Research Fund with Grant No. 044/2565.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the manuscript.

Acknowledgments: The authors and each project member wish to gratefully acknowledge financial support for this research from the Srinakharinwirot University Research fund.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Escobedo, E.; Ramirez, L.; Camara, G. Dynamic sign language recognition based on convolutional neural networks and texture maps. In Proceedings of the 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Rio de Janeiro, Brazil, 28–30 October 2019; pp. 265–272. [\[CrossRef\]](#)
- Cheok, M.J.; Omar, Z.; Jaward, M.H. A review of hand gesture and sign language recognition techniques. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 131–153. [\[CrossRef\]](#)
- Al-Hammadi, M.; Bencherif, M.A.; Alsulaiman, M.; Muhammad, G.; Mekhtiche, M.A.; Abdul, W.; Alohal, Y.A.; Alrayes, T.S.; Mathkour, H.; Faisal, M.; et al. Spatial attention-based 3D graph convolutional neural network for sign language recognition. *Sensors* **2022**, *22*, 4558. [\[CrossRef\]](#) [\[PubMed\]](#)
- Pradeep, A.; Asrorov, M.; Quronboyeva, M. Advancement of Sign Language Recognition through Technology Using Python and OpenCV. In Proceedings of the 2023 7th International Multi-Topic ICT Conference (IMTIC), Jamshoro, Pakistan, 10–12 May 2023; pp. 1–7. [\[CrossRef\]](#)
- Stefanov, K.; Beskow, J. A Real-time Gesture Recognition System for Isolated Swedish Sign Language Signs. In Proceedings of the 4th European and 7th Nordic Symposium on Multimodal Communication (MMSYM), Copenhagen, Denmark, 29–30 September 2016; pp. 18–27.
- Kudrinko, K.; Flavin, E.; Zhu, X.; Li, Q. Wearable sensor-based sign language recognition: A comprehensive review. *IEEE Rev. Biomed. Eng.* **2020**, *14*, 82–97. [\[CrossRef\]](#) [\[PubMed\]](#)
- Cheng, J.; Wei, F.; Liu, Y.; Li, C.; Chen, Q.; Chen, X. Chinese sign language recognition based on DTW-distance-mapping features. *Math. Probl. Eng.* **2020**, *2020*, 8953670. [\[CrossRef\]](#)
- Saggio, G.; Calado, A.; Errico, V.; Lin, B.S.; Lee, I.J. Dynamic Measurement Assessments of Sensory Gloves Based on Resistive Flex Sensors and Inertial Measurement Units. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 9505410. [\[CrossRef\]](#)
- Rastgoo, R.; Kiani, K.; Escalera, S. Hand pose aware multimodal isolated sign language recognition. *Multimed. Tools Appl.* **2021**, *80*, 127–163. [\[CrossRef\]](#)
- Xu, B.; Huang, S.; Ye, Z. Application of tensor train decomposition in S2VT model for sign language recognition. *IEEE Access* **2021**, *9*, 35646–35653. [\[CrossRef\]](#)
- Goswami, T.; Javaji, S.R. CNN model for american sign language recognition. In Proceedings of the ICCCE 2020, Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering, Paris, France, 18–20 July 2021; Springer: Singapore, 2021; pp. 51–61.
- Uyyala, P. Sign language recognition using convolutional neural networks. *J. Interdiscip. Cycle Res.* **2022**, *14*, 1198–1207. [\[CrossRef\]](#)
- Yuan, G.; Liu, X.; Yan, Q.; Qiao, S.; Wang, Z.; Yuan, L. Hand gesture recognition using deep feature fusion network based on wearable sensors. *IEEE Sens. J.* **2020**, *21*, 539–547. [\[CrossRef\]](#)
- Borg, M.; Camilleri, K.P. Phonologically-meaningful subunits for deep learning-based sign language recognition. In Proceedings of the Computer Vision—ECCV 2020 Workshops, Glasgow, UK, 23–28 August 2020; Proceedings, Part II 16. pp. 199–217.
- Rusu, A.A.; Rabinowitz, N.C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. Progressive neural networks. *arXiv* **2022**, arXiv:1606.04671. [\[CrossRef\]](#)
- Li, C.; Zhuang, B.; Wang, G.; Liang, X.; Chang, X.; Yang, Y. Automated progressive learning for efficient training of vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12486–12496.
- Rastgoo, R.; Kiani, K.; Escalera, S. Real-time isolated hand sign language recognition using deep networks and SVD. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 591–611. [\[CrossRef\]](#)
- Al-Hammadi, M.; Muhammad, G.; Abdul, W.; Alsulaiman, M.; Bencherif, M.A.; Alrayes, T.S.; Mathkour, H.; Mekhtiche, M.A. Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation. *IEEE Access* **2020**, *8*, 192527–192542. [\[CrossRef\]](#)

19. Aly, S.; Aly, W. DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition. *IEEE Access* **2020**, *8*, 83199–83212. [[CrossRef](#)]
20. Li, D.; Rodriguez, C.; Yu, X.; Li, H. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1459–1469.
21. DelPreto, J.; Hughes, J.; D’Aria, M.; de Fazio, M.; Rus, D. A Wearable Smart Glove and Its Application of Pose and Gesture Detection to Sign Language Classification. *IEEE Robot. Autom. Lett.* **2022**, *7*, 10589–10596. [[CrossRef](#)]
22. Rosero-Montalvo, P.D.; Godoy-Trujillo, P.; Flores-Bosmediano, E.; Carrascal-García, J.; Otero-Potosi, S.; Benitez-Pereira, H.; Peluffo-Ordóñez, D.H. Sign language recognition based on intelligent glove using machine learning techniques. In Proceedings of the 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, 15–19 October 2018; pp. 1–5. [[CrossRef](#)]
23. Zhang, Q.; Wang, D.; Zhao, R.; Yu, Y. MyoSign: Enabling end-to-end sign language recognition with wearables. In Proceedings of the 24th International Conference on Intelligent User Interfaces, Marina del Ray, CA, USA, 17–20 March 2019; pp. 650–660. [[CrossRef](#)]
24. Nandi, U.; Ghorai, A.; Singh, M.M.; Changdar, C.; Bhakta, S.; Kumar Pal, R. Indian sign language alphabet recognition system using CNN with diffGrad optimizer and stochastic pooling. *Multimed. Tools Appl.* **2023**, *82*, 9627–9648. [[CrossRef](#)]
25. Rwelli, R.E.; Shahin, O.R.; Taloba, A.I. Gesture based Arabic Sign Language Recognition for Impaired People based on Convolution Neural Network. *arXiv* **2022**, arXiv:2203.05602. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.