sensors

MDPI

*Article*

# American Sign Language Words Recognition of Skeletal Videos Using Processed Video Driven Multi-Stacked Deep LSTM

**Sunusi Bala Abdullahi** [1,2,†] and **Kosin Chamnongthai** [3,*,†]

[1] Department of Computer Engineering, Faculty of Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand; sbabdullahi@ieee.org

[2] Zonal Criminal Investigation Department, The Nigeria Police, Louis Edet House Force Headquarters, Shehu Shagari Way, Abuja 900221, Nigeria

[3] Department of Electronic and Telecommunication Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand

* Correspondence: kosin.cha@kmutt.ac.th

† These authors contributed equally to this work.

**Abstract:** Complex hand gesture interactions among dynamic sign words may lead to misclassification, which affects the recognition accuracy of the ubiquitous sign language recognition system. This paper proposes to augment the feature vector of dynamic sign words with knowledge of hand dynamics as a proxy and classify dynamic sign words using motion patterns based on the extracted feature vector. In this method, some double-hand dynamic sign words have ambiguous or similar features across a hand motion trajectory, which leads to classification errors. Thus, the similar/ambiguous hand motion trajectory is determined based on the approximation of a probability density function over a time frame. Then, the extracted features are enhanced by transformation using maximal information correlation. These enhanced features of 3D skeletal videos captured by a leap motion controller are fed as a state transition pattern to a classifier for sign word classification. To evaluate the performance of the proposed method, an experiment is performed with 10 participants on 40 double hands dynamic ASL words, which reveals 97.98% accuracy. The method is further developed on challenging ASL, SHREC, and LMDHG data sets and outperforms conventional methods by 1.47%, 1.56%, and 0.37%, respectively.

**Keywords:** American sign language words; bidirectional long short-term memory; computer vision; deep learning; dynamic hand gestures; leap motion controller sensor; sign language recognition, ubiquitous system; video processing

check for updates

**Citation:** Abdullahi, S.B.; Chamnongthai, K. American Sign Language Words Recognition of Skeletal Videos Using Processed Video Driven Multi-Stacked Deep LSTM. *Sensors* **2022**, *22*, 1406. https://doi.org/10.3390/s22041406

Academic Editor: José Luis Alba Castro

Received: 13 January 2022
Accepted: 8 February 2022
Published: 11 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2022 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

Among sign languages, which are normally used in deaf communication, American sign language (ASL) is one of the standard [1–3] and popularly used sign language across the world. ASL words are performed using single and double hands in the deaf communication, and majority of ASL words are performed using double hands, which are dominant and non-dominant hands [4,5]. Several single-handed words have now added a second hand in an identical or reciprocal rotation, to increase redundancy. Such redundancy is a significant parameter in sign to discriminate similarity and to predict other parameters [6]. These double-hand sign words share some similar features, which usually occur at the beginning and ending of sign trajectory, which leads to misunderstandings. Most double hand sign words are dynamic words. Classification of dynamic sign words using single and double hands is the basic function for automatic sign language recognition applications; especially, the recognition of similar double hand sign words is an important and useful research problem in terms of accuracy.

It is observed from available existing works that sign words recognition has been performed using single or double hands and can be classified into deep learning and feature-based systems, as illustrated in Table 1. The deep learning approach recently emerged and can deal excellently with problems related to big data [7–11]. However, some problems may not be convenient for collecting a huge number of samples such as big data; thus, a feature-based system is introduced, in which users consider features in advance, and it performs well in some cases. The feature based-system for sign word recognition can be categorized into two groups: static and dynamic sign word groups. Research works in static sign words recognition [12,13] can recognize sign words to a high degree of accuracy but may not work well with dynamic sign words since the majority of dynamic sign words express their meaning in motion. Research works in dynamic sign word recognition are grouped into single- and double-hand dynamic sign words. Works in single-hand [5,14–17] dynamic sign words are limited due to the fact most of the sign letters or alphabets are less complex. Double-hand sign words are commonly used in daily communications, but complex hand motion interactions are major challenge. Recognition of double-hand both static and dynamic Indian gestures is proposed in [18]. The authors developed a system of utilizing feature points engineered from a minimum Eigen value algorithm to recognize double-hand, which was later converted using COM server in MATLAB as both text and speech. The method is limited to only statistical features. Demrcioglu et al. [19] designed a double-hand sign words recognition system from hand shapes, using three machine learning classifiers, among which the heuristics classifier outperformed others with good recognition accuracy, but the method suffers from insufficient features representation since the majority of double-hand actions are characterized by hand shape, motion and orientation. The work in Deriche [20] proposed the CyberGlove with the SVM model for double-hand dynamic word recognition. This method achieved good recognition; however, Cyberglove for sign words recognition is expensive, intrusive, and has imprecise calibrations. Haque et al. [21] designed a Two-Handed Bangla Sign Language Recognition system that recognizes 26 sign gestures, from a three-structured flow. Their method extracts images using Principal Component Analysis (PCA) and K-Nearest Neighbors (k-NN), which are used as classification algorithms. This method achieved a success rate of 77.8846% by testing 104 images. The major disadvantage of this method is low accuracy because of a complex camera background, as well as a limited number of considered features, while Raghuveera et al. [22] proposed ensemble learning using SVM from SURF, HOG and LBP hand features, to control complex camera background. This method achieved low accuracy because of non-scalable features, similarity, and complex segmentation. Karaci et al. [23] presented ASL letters using LR, k-NN, RF, DNN, and ANN classifiers for double-hand sign language recognition. The overall result of these experiments using cascade voting achieved an accuracy of 98.97%. The system can be useful for finger-spelling/letter signs only. The ASL recognizer system cannot be considered as a complete SLR system because we have to include dynamic sign words. Meanwhile, ref. [24] developed a system for Turkish dynamic sign word recognition based on multi-layer kernel-based extreme learning machine (ML-ELM) algorithm. The proposed method was capable of successfully recognizing sign words in the dictionary with an accuracy of 98%. The primary disadvantage of using ML-KELM is the problem of obtaining a least square optimal solution of ELM. A double-hand SLR application system using LMC with a Windows platform is proposed as an expert system in [25]. Hisham and Hamouda [26] built double and single-hand sign words recognition inside Latte Panda with an Ada-Boosting strategy. The method achieved good recognition but cannot learn sequential data and may fail with complex sign words. Researchers identified the potential of using recurrent neural network and its variants to effectively learn long-term dependencies for sign language recognition [27–29]. However, single LSTM has weak learning ability [30,31], and it falls easily into over-fitting, in contrast to multi-LSTM network [29]. A similarity problem of double-hand dynamic sign words is addressed by Avola et al. [32] using multi-stacked LSTM learning; they utilized 3D hand internal angles, position displacements of the palm, and the fingertips Equation (5) to recognize dynamic ASL words [32]. The work in [32] is good at recognizing some dynamic ASL words, but the major disadvantage of achieving large abstraction (deeper network) via the multi-stacked LSTM method is that learning ability is marginal when the sample feature is increased, and, consequently, the recognition rate does not significantly improve. However, ref. [32] considers a limited number of ASL dynamic words, and their handcrafted features are not sufficient to recognize most available dynamic ASL words, especially sign words from similar class. Thus, these models/algorithms are insensitive to human hand dynamics and cannot use various classes of features, which leads to bad extensibility. These two problems may lead to misclassification of double-hand dynamic ASL words. We observed that the existing methods failed to utilize about 7% of the first few video frames during segmentation. These discarded frames contain a hand pause feature, which is not properly processed by the existing recognition methods.

For this reason, we propose to utilize the 3D extended kalman filter (EKF) covariance matrix feature representation of double-hand motion trajectories and to add a hand pause feature, as our feature vector for double-hand dynamic sign words recognition. Skeletal videos from LMC are affected by noise, and we deploy a robust weighted least square (WLS) algorithm where each sequence is allocated with effective weights to obtain the best confidence score with the fewest residuals. The corrected video sequences are fed into the EKF to track 3D double-hand motion trajectories across video frames through estimating anonymous features by approximating a probability density function over the entire video sequence. Basic hand features (hand shape, orientation, position and motion) are automatically extracted from skeletal hand-joint videos using bi-directional recurrent neural network (BiRNN). The extracted features are transforms using maximal information correlation (MIC) and rows concatenation for best feature representation. Finally, the selected features are computed using video frames correction to control initial frame coordinates and positions. To this end, we design a consolidated feature vector to achieve flexible and effective learning of double-hand complex gesture recognition. Moreover, none of the existing literature has tried to use the performance of networks to optimize loss function. This paper intended to bridge this gap. In addition to the mentioned research focus, dynamic hand gesture recording and recognition was applied in various consumer applications [33–35]. We made the following contributions:

(a) Acquisition and processing of skeletal video images acquired by means of a portable leap motion controller (LMC) sensor.

(b) The development of an EKF-tracking to address hand motion tracking errors and uncertainties across each frame in obtaining hand motion trajectories.

(c) The development of an innovative algorithm based on WLS to control noise across video frames.

(d) The design of a BiRNN network that is able to extract the proposed features from raw skeletal video frames.

(e) The development of an MIC scheme to select the most significant features from raw video images. These are used as input to the multi-stacked deep BiLSTM recognition network to discriminate among similar double-hand dynamic ASL words.

(f) Intensive evaluation using Jaccard, Mathew correlation and Fowlkes–Mallows indices is carried out to analyze the reliability of recognition results. These indices estimate the confusion matrix via known parameters for assessing the probability that the performance would be achieved by chance, due to the assumption of randomness of the k-fold and LOSO cross-validation protocol.

(f) Investigation of the best recognition network by comparing the performance of Adam, AdaGrad and Stochastic gradient descent on loss function, for ubiquitous applications.

The remainder of our article is structured as follows: Section 1 provides relevant works; Section 2 provides basic feature definitions, skeletal video preprocessing, WLS, hand tracking using EKF, MIC, features-scaling, skeletal-video-frames correction, ASL words recognition from skeletal video feature, BiRNN features extraction, LSTM, model parameters, evaluation metrics, experiments, and data set design; Section 3 provides results and details of a performance comparison with baseline methods; Section 4 discusses the implemented approach; and Section 5 concludes the entire work.

**Table 1.** Table of Related works.

| Algorithm Name | Methodology | Results (%) | Limitations |
|---|---|---|---|
| 1. DEEP LEARNING-BASED SYSTEM | | | |
| Konstantinidis et al. [8] | Meta-learner + stacked LSTMs | 99.84 and 69.33 | Computational complexity |
| Ye et al. [10] | 3DRCNN + FC-RNN | 69.2 for 27 ASL words | annotation and labeling is required |
| Parelli et al. [36] | Attention-based CNN | 94.56 and 91.38 | large-scale data set required |
| Asl-3dcnn [11] | cascaded 3-D CNN | 96.0, 97.1, 96.4 | spatial transformation |
| B3D ResNet [7] | 3D-ResConVNet + BLSTM | 89.8 and 86.9 | Misclassification + large data require |
| Rastgoo et al. [9] | SSD + 2DCNN + 3DCNN + LSTM | 99.80 and 91.12 | complex background + large data |
| 2. FEATURE-BASED SYSTEM | | | |
| 2.1 Static sign words | | | |
| Mohandes et al. [12] | MLP + Naïve Bayes | 98 and 99 | Not very useful for daily interact |
| Nguyen and Do [13] | HOG + LBP + SVM | 98.36 | Not very useful for daily interact |
| 2.2 Dynamic Sign words | | | |
| 2.2.1 single-hand dynamic sign words recognition | | | |
| Naglot and Kulkarni [14] | LMC + MLP | 96.15 | Misclassification |
| Chopuk et al. [16] | LMC + polygon region + Decision tree | 96.1 | Misclassfication |
| Chong and Lee [15] | LMC + SVM + DNN | 80.30 and 93.81 | Occlusion and similarity |
| Shin et al. [17] | SVM + GBM | Massey 99.39,<br>ASL alphabet 87.60<br>FingerspellingA 98.45. | error in estimated 3D coordinates |
| Vaitkevicius et al. [37] | SOAP + MS SQL + HMC | 86.1 ± 8.2 | Misclassfication |
| Lee et al. [5] | LSTM + k-NN | 99.44, at 5-fold 91.82 | limited extensibility due to tracking |
| Chophuk and Kosin [31] | BiLSTM | 96.07 | limited extensibility to double hand |
| 2.2.2 double-hand dynamic sign words recognition | | | |
| Igari and Fukumura [38] | minimum jerk trajectory + DP-matching + Via-points + CC | 98 | Cumbersome + limited number of features |
| Dutta and GS [18] | Minimum EigenValue + COM Server | Text + Speech | Poor extensibility to word |
| Demrcioglu et al. [19] | Heuristics + RF + MLP | 99.03, 93.59, and 96.67 | insufficient hand features |
| Deriche [20] | CyberGlove + SVM | 99.6 | Cumbersome + intrusive |
| DLMC-based ArSLRs [39] | LDA + GMM bayes classifier | 94.63 | sensor fusion complexity<br>separate feature learning |
| Deriche et al. [40] | Dempster-Shaper + LDA + GMM | 92 | Misclassification and not mobile |
| Haque et al. [21] | Eigenmage + PCA + k-NN | 77.8846 | complex segmentation + few feature |
| Raghuveera et al. [22] | SURF + HOG + LBP +SVM | 71.85 | non-scalable features + segmentaton |
| Mittal et al. [30] | CNN + LSTM | Word 89.5 | low accuracy due to weak learning |
| Katilmis and Karakuzu [41] | LDA + SVM + RF | 93, 95 and 98 | letters are not very useful for daily communication |
| Karaci et al. [23] | LR + k-NN + RF + DNN + ANN | cascade voting achieve 98.97 | letters are not very useful for daily<br>Fails to track double hands |
| Kam and Kose [25] | Expert systems + LMC + WinApp | SLR App | Mobility is not actualized |
| Katilmis and Karakuzu [24] | ELM + ML-KELM | 96 and 99 | complex feature extension may fall into over-fitting |
| Hisham and Hamouda [26] | Latte Panda + Ada-Boosting | double hand accuracy 93 | similarity due to tracking issues |
| Avola et al. [32] | Multi-stacked LSTM | 96 | insufficient hand features |

## 2. Materials and Methods

This section enumerates double-hand dynamic ASL words sign language recognition processes of the proposed method. We introduce our method in four subsections as follows: Section 2.2 skeletal video preprocessing, which encompasses the following: (a) weighted least square (WLS) algorithm for minimizing noise of 3D skeletal video sequence, (b) hand tracking using EKF method for tracking deep hand motion trajectories across video frames, (c) MIC for robust features selection, (d) features scaling to control hand dynamics and allow new signer, and (e) skeletal video frames correction to control initial frame coordinates and position of all consecutive frames. Section 2.3 ASL words recognition from skeletal video features encompasses the following stages: (a) bidirectional RNN (BiRNN) features extraction, (b) long short-term memory, and (c) multi-stacked deep BiLSTM training from transfer learning to learn temporal continuity of dynamic words. Section 2.4 encompasses model parameters. Section 2.5 encompasses evaluation metrics to calculate the overlap and similarity among the original dynamic ASL words and the predicted category videos for the recorded ASL words. Overall procedures of the adopted method are shown in Figure 1.

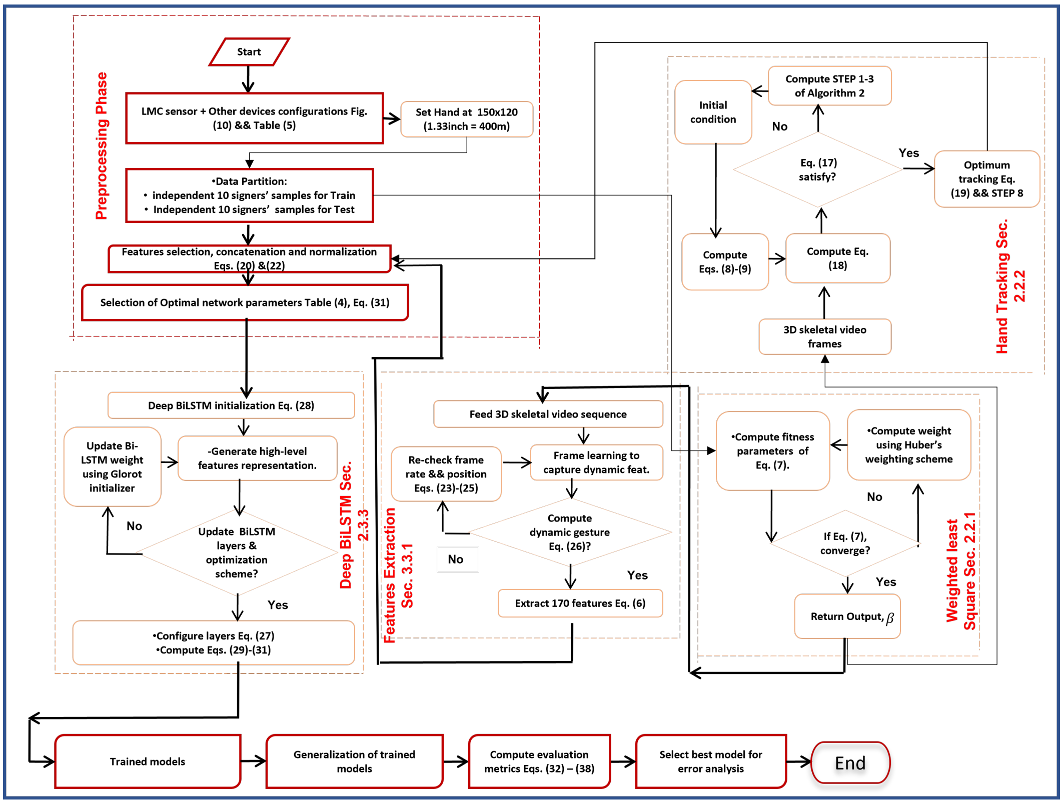

Preprocessing Phase
Start
LMC sensor + Other devices configurations Fig. (10) && Table (5)
Set Hand at 150x120 (1.33inch = 400m)
•Data Partition:
• independent 10 signers' samples for Train
• Independent 10 signers' samples for Test
Features selection, concatenation and normalization Eqs. (20) &(22)
Selection of Optimal network parameters Table (4), Eq. (31)
Compute STEP 1-3 of Algorithm 2
Initial condition
No
Eq. (17) satisfy?
Yes
Optimum tracking Eq. (19) && STEP 8
Compute Eqs. (8)-(9)
Compute Eq. (18)
3D skeletal video frames
Hand Tracking Sec. 2.2.2
Deep BiLSTM initialization Eq. (28)
Update Bi-LSTM weight using Glorot initializer
-Generate high-level features representation.
No
Update BiLSTM layers & optimization scheme?
Yes
•Configure layers Eq. (27)
•Compute Eqs. (29)-(31)
Deep BiLSTM Sec. 2.3.3
Feed 3D skeletal video sequence
Re-check frame rate && position Eqs. (23)-(25)
Frame learning to capture dynamic feat.
Compute dynamic gesture Eq. (26)?
No
Yes
Extract 170 features Eq. (6)
Features Extraction Sec. 3.3.1
•Compute fitness parameters of Eq. (7).
•Compute weight using Huber's weighting scheme
No
If Eq. (7), converge?
Yes
Return Output, $\beta$
Weighted least Square Sec. 2.2.1
Trained models
Generalization of trained models
Compute evaluation metrics Eqs. (32) – (38)
Select best model for error analysis
End

**Figure 1.** Procedures of the proposed method.

### 2.1. Basic Definition of Multi-Stacked Deep BiLSTM Feature

Sign language basic features (phonemes) [42] include hand shape, motion, orientation and location. (1) 3D dynamic hand shape characterizes double-hand dynamic ASL words, which can be obtained from the twenty-two skeletal hand joint primes $L$ per each hand, thus, making a total of 44 primes for the double-hand $L_{44}$, along seventy angular features $\omega_{70}$ for the complete double-hand as described in Figure 2 and put in Equation (2). (2) 3D hand orientation provides angle coordinates at which the double-hands meet each other. The hand orientation angle is computed from seventy angle primes of seven major double-hand vertices, as described in Figure 2. However, hand location/position is obtained from direct measurement using LMC. Deep features are defined differently, but for the purpose of this article we have considered the following deep features. (1) Double-hand motion trajectories (MT), while performing ASL word, are defined as the action of two-hands in the LMC sensor's field of view. This action is visualized as trajectory across video frames Equation (3) and can be tracked based on EKF algorithm. MT encodes correlation among hand movement and gesture dynamics. MT allow one to learn each dynamic across frames and to observe points where two gestures share similar characteristics, as mathematically established in Equation (3). Hand motion usually determines the frame speed of the video, which is coined in Equation (4). 3D dynamic hand motion is composed of velocity, which is comprises of action at beginning of gesture performance (preparation), peak acceleration (nucleus), and ending of gesture performance (retraction). Beginning and ending of gesture trajectory are known as preparation and retraction (that is, pause). (2) Hand Pause provides another potential information to discriminate similarity between dynamic gesture at the beginning or end of gesture characteristics. Thus, hand pause $P$ is mathematically formulated within the leap motion Euclidean space in Equation (1). Significance of the proposed features to recognize double-hand ASL words is investigated using maximal information criterion (MIC) and cumulative match characteristics (CMC) curve.

$$||P(t)|| = \sqrt{P_p^{(2)}(t) + P_q^{(2)}(t) + P_r^{(2)}(t)} \tag{1}$$

where $P(t) = (p, q, r) \in R^3$.

$$\eta(p, q, r) = (L_{44} + \omega_{70}). \tag{2}$$

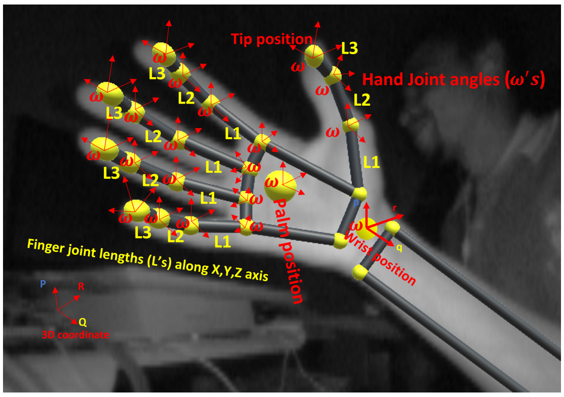

Tip position
Hand Joint angles ($\omega$'s)
L3
L2
L1
$\omega$
Palm position
Wrist position
Finger joint lengths (L's) along X,Y,Z axis
P
R
Q
3D coordinate

**Figure 2.** Skeletal palm joints with angle and joint length primes captured by LMC [43].

However, for each dominant hand in video frame $f$ at time $t$, while moving towards non-dominant hand (that is, the hands lined up to their orientation), hand motion trajectories across the consecutive frames at time $T$, can be expressed as $[M_t^{(f)}]_{t=1,\cdots,T}$, where $M_t^f$ is defined as:

$$M_t^{(f)}(p, q, r) = (p_t^{(f)} + \sin\varphi_t^{(f)}, q_t^{(f)} + \cos\varphi_t^{(f)}, r_t^{(f)} + \tan\varphi_t^{(f)}) \in R^3. \tag{3}$$

ASL word motion speed trajectory $K_t$ can be obtained at each fingertip. The fingertips provide hand motion in Equation (3), which can be formulated as follows:

$$K_t(p, q, r) = ||k_f - k_{f-1}|| \tag{4}$$

where motion variation from $f$th to frame $f$th + 1 denotes speed difference and its correlation. With the addition of $(P_t^f)$, $(\eta_t)$ and $(K_t)$ features, the functional Equation (6) can improve accuracy and reduce misclassification from double hand similar ASL words. Finally, the proposed features vector $(\beta)$ of model [32] is defined by:

$$\beta_t(p, q, r) = [\omega_t^f, N_t] \tag{5}$$

To improve recognition accuracy and minimize misclassification of a set of double-hand dynamic ASL word feature vector, new features $(P_t^f)$, $(\eta_t)$ and $(K_t)$, called basic and deep features, are added in Equation (5), and their functions are discussed in Equations (1)–(4), which can be uniformly written as:

$$\beta_t(p, q, r) = (P_t^f + \omega_t^f + \eta_t + N_t + K_t + \varphi_t) \tag{6}$$

where $(P_t^f)$, $(\omega_t^f)$, $(\eta_t)$, $(N_t)$, $(K_t)$ and $(\varphi_t)$ denote pause, angles, shapes, positions (palm position displacement and fingertip position displacements), motion, and relative trajectory features in frame $f$th, at time $t$, respectively. Relative trajectory includes hand motion trajectories, speed, and relationship between dominant and non-dominant hand.

### *2.2. Skeletal Video Preprocessing*

Noise such as large video frame sizes (due to large recording time) and human hand dynamics affects recognition performance of double-hand dynamic skeletal video information. The following sections employ robust tools to preserve the original video information free from noise.

### 2.2.1. Weighted Least Square (WLS)

Skeletal video sequences are affected by noise (missing values), which has detrimental effect during recognition. This noise information is manifested among different video sequences, which influence the estimated original video sequence. To address this problem, WLS algorithm is chosen. WLS overcomes traditional drawback of linear regression, moving average and median filter problem of filtering only data sets with constant variance. WLS is a good choice of filter for many researchers in video processing [44–48]. Therefore, each sequence is allocated with suitable and effective weights to achieve significant confidence level with least residuals. The minimization of the errors in WLS is iteratively learned until weights of outliers are minuscule. The weights are obtained using Huber's weighting scheme [44]. A given ($A$), ($A^T$), ($O$) and ($D$), which denote weight matrix, matrix transpose, response vector and diagonal matrix, contains weights associated with video samples; then, $\tilde{\beta}$ returns the estimate, as explained in Algorithm 1. WLS can be formulated as follows:

$$(A^T DA(w))\lambda = A^T * D * O \tag{7}$$

where $w$, $c$, $\beta(w)$, $w_f$, $d(f)$ and $l_f$ denote prediction time, order of prediction, raw video information, video progressing time, filter input and linear function. $\lambda$ denotes wavelength parameter.

**Algorithm 1:** (*WLS*).

```
***Input.*** Set $l_f(w) = [100 \cdots 0]^c$, $d(w)$
***Output.*** WLS estimate $\tilde{\beta} = (A^T DA(w))\lambda$
***Step 1.*** Compute $\beta(w) = l_f^c(w-1)d(w)$
***Step 2.*** Compute $l_f(w)$
***Step 3.*** Compute $A(w_f)$ in Equation (7)
***Step 4.*** Update $A(w_f)$ then
***Step 5.*** Finally we set $w := w + 1$ and return to step 1.
```

### 2.2.2. Hand Tracking Using Kalman Extended Filter (EKF)

The EKF is computationally efficient to our proposed data set, and the brief process is illustrated in flow chart Figure 1 and Algorithm 2. In each video frame, the two skeletal hands are learned from their registered starting point ($P_t$) to the hand resting point ($P_{t+1}$) while recording, as illustrated in Figure 3. The EKF involves estimating the process state with the aid of the equation of partial derivative and observation, using equation of partial derivative of process and observation, as shown in Equations (13) and (14). In Equation (13), $s \in R^\iota$, $\vartheta$, $s_c$ and $\zeta_c$ denote nonlinear function, state variable, and process noise (feed back from LMC sensor). The nonlinear function evaluates the state according to the current moment $c$. The function parameters will extrapolate $g_{c-1}$ and $\zeta_c$. In Equation (14), $d \in R^\tau$, $d_c$, $\phi$ and $\Omega_c$ denote observed variable, nonlinear function and observation noise (feed back from LMC sensor). Therefore, to incorporate the process of a nonlinear difference and observation noise for real-life usage, modified Equations (13) and (14) are adopted from [49]:

$$s_c \approx \tilde{s_c} + I(s_{c-1} - \hat{s}_{c-1}) + \zeta\zeta_{c-1} \tag{8}$$

where $s_c$, $I$, $\tilde{s_c}$ and $\hat{s}_{c-1}$ denote original information of the state vector, Jacobian matrix of the partial derivative of $\vartheta$ with respect to $s$, observation information of the state vector, and state vector posteriori probability of moment $c$.

$$d_c \approx \tilde{d_c} + U(s_c - \tilde{s_c}) + \Omega\Omega_c \tag{9}$$

where $d_c$, $U$ and $\tilde{d_c}$ denote original information of the observation vector, Jacobian matrix of the partial derivative of $\phi$ with respect to $\Omega$, and observation information of the state vector. The Jacobian cannot be estimated mathematically with noise term; therefore, it is assumed as zero. Thus, the Jacobian matrices can be obtained as follows:

$$I_{n,f} = \frac{\partial \vartheta_n}{\partial s_f}\{\hat{s}_{c-1}, g_{c-1}, 0\} \tag{10}$$

$$U_{n,f} = \frac{\partial \Omega_n}{\partial s_f}\{\tilde{s}_c, 0\}. \tag{11}$$

However, the residuals of the observed variables and prediction error can be obtained from the covariance matrix in Equation (19).

The covariance matrix is independent from random variables that provide an approximation using Equations (13) and (14). From this approximation, the EKF can be extended to estimate the equation, thus

$$\hat{s}_c = \tilde{s}_c + C_c\tilde{r}_{d_c} = \tilde{s}_c + C_c(d_c - \tilde{d}_c). \tag{12}$$

Finally, Equation (12) is utilized to adopt the observation variables of EKF. $\hat{s}_c$ and $\tilde{d}_c$ can be obtained from Equations (13) and (14), respectively. From the results in Figure 3, we have the following observations. (1) Blue plot indicates the original 3D hand motion trajectory along with its corresponding mean square error (MSE). Red plot indicates the estimated 3D hand motion trajectory along with its corresponding MSE. Individual axis performance of EKF algorithm is demonstrated by the left plots. EKF algorithm achieves very competitive tracking across the 3-axis by observing the MSE, which validates the stability of EKF algorithm for complete hand motion trajectory. (2) As the ambiguity/uncertainty rate increases, the performance degradation (high MSE) of the compared original measurements is much larger than that of EKF tracking.

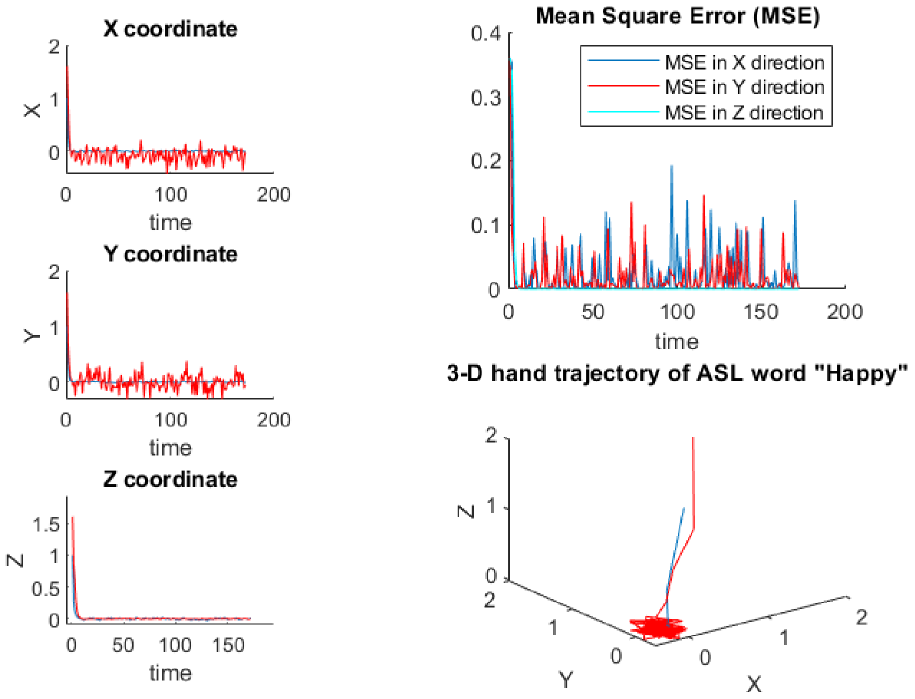


**Figure 3.** 3D Hand motion trajectories across video frames using EKF.

**Algorithm 2:** (EKF).

```
Input. Choose any arbitrary actual initial conditions $w$, initial observations $m$, assumed
  initial conditions $j$, covariance of estimation initial value $h$, the sampling time $t$, $indx = 0$,
  and $n = 1:170$.
Initial setting. Let $d_c$, $s_c$, $h$ and $S$ be covariance matrix of process noise, measurement
noise, estimation error and original information.
Output. 3D EKF estimate $\tilde{M}_n$.
Step 1. Determine process and observation along X, Y, and Z coordinates, from
        $s_c = \vartheta\{s_{c-1}, g_{c-1}, \zeta_{c-1}\}$        (13)
        $d_c \in \phi\{s_c, \Omega_c\}$        (14)
Step 2. Compute prediction function
        $j(n), h(n) := predict(S, j(n), h(n), d_c).$        (15)
Step 3. Compute Jacobian matrices in Equations (10) and (11).
Step 4. Computes Kalman gain
        $EKF(n+1) = Gain(H(n+1), P(n+1), M)$        (16)
Step 5. Compute overall estimate
        $j(n+1) = j(n+1) + EKF(n+1) * G$        (17)
where
Step 6. G is the filter specialty, estimates from
        $G = G(m(n+1), j(n+1), indx)$        (18)
Step 7. Compute covariance estimation error
        $\hat{s}_c = \tilde{s}_c + \hat{r}_c$        (19)
Step 8. Compute MSE along X, Y, and Z. as shown in Figure 3
Step 9. Finally, we set $n := n + 1$ and return to step 1.
```

#### 2.2.3. Maximal Information Correlation (MIC)

We introduced a feature selection method derived from correlation analysis known as MIC to reduce the complexity of the deep learning algorithms. MIC utilizes 3D video features between zero and one. The significance of adopting MIC was the capacity to treat nonlinear and linear unions among video data sets. It makes no assumptions about the distribution of the recorded video. However, MIC has simple computing formula, and it applies to sample sizes $t \geq 2$. MIC of 3D vectors $p$, $q$ and $r$ is defined as follows [50,51], and the results are displays in Table 2:

$$MIC = max\{\frac{I(p,q,r)}{log_2 min(t_p, t_q, t_r)}\} \tag{20}$$

where

$$\begin{aligned} I(p,q,r) = H(p) + H(q) + H(r) - H(p,q,r) = \sum_{u=1}^{t_p} p(p_u) log_2 \frac{1}{p(p_u)} + \sum_{v=1}^{t_q} p(q_v) log_2 \frac{1}{p(q_v)} \\ + \sum_{w=1}^{t_r} p(r_w) log_2 \frac{1}{p(r_w)} - \sum_{u=1}^{t_p} \sum_{v=1}^{t_q} \sum_{w=1}^{t_r} p(p_u, q_v, r_w) log_2 \frac{1}{p(p_u, q_v, r_w)} \end{aligned} \tag{21}$$

where $p$, $q$ and $r$ denote feature vectors along 3D axis. $H$; $I$; $B$; and $p_u$, $q_v$ and $r_w$ denote entropy, information, bins and number of bins of the partition along 3D axis. Note:

$p_u.q_v.r_w < B(t)$ and $B(t) = t^{(0.6)}$. The MIC analysis demonstrates the effectiveness of the proposed features as shown in Table 2. In Table 2, the diagonal values indicate correlation of each feature with itself, while all other values inside the table indicate the correlation of each feature against it neighbor. Positions having values ranges 0.9 to 1 are regarded as having strong correlation, whereas values less than 0.9 are still significant and are conserved during model design. All other features less than 0.8 were disregarded in this paper. Furthermore, we investigated the significance of the selected features according to the cumulative match characteristics curve (CMC), as illustrated in Figure 4. The CMC plot is generally used to quantify the correlation between detection rate and the rank score from the given features. We evaluated different feature combinations across all the features, but the following were found to be effective according to CMC ranking: 1st (shape + position + motion), 2nd (shape + position + angle), 3rd (shape + position + motion + angle), 4th (shape + angle), 5th (shape + position + motion + angle + pause + relative trajectory) and 6th (shape + position). In this plot, each features combination exempted the knowledge of hand dynamics (pause and relative trajectory), while the remaining features were evaluated so that measure of the contribution of our added feature per each combinations was obtained. Thus, best feature combinations were achieved with least score at 5th rank (shape + positions + motion + angles + pause + relative trajectory features), whereas less significant features combination was achieved with high score from the 6th rank (shape + position features), as shown in Figure 4. Therefore, it is difficult to achieve best recognition with features combination, due to absence of hand dynamics knowledge.

**Table 2.** Results of correlational analysis.

| | Shape | Motion | Position | Angles | Pause | Relative Trajectories |
|---|---|---|---|---|---|---|
| Shape | 1 | | | | | |
| Motion | 0.9444 | 1 | | | | |
| Position | 0.8781 | 0.9351 | 1 | | | |
| Angles | 0.86728 | 0.93985 | 0.84453 | 1 | | |
| Pause | 0.87361 | 0.71931 | 0.90719 | 0.89857 | 1 | |
| Relative trajectories | 0.95351 | 0.94203 | 0.89075 | 0.90681 | 0.81375 | 1 |

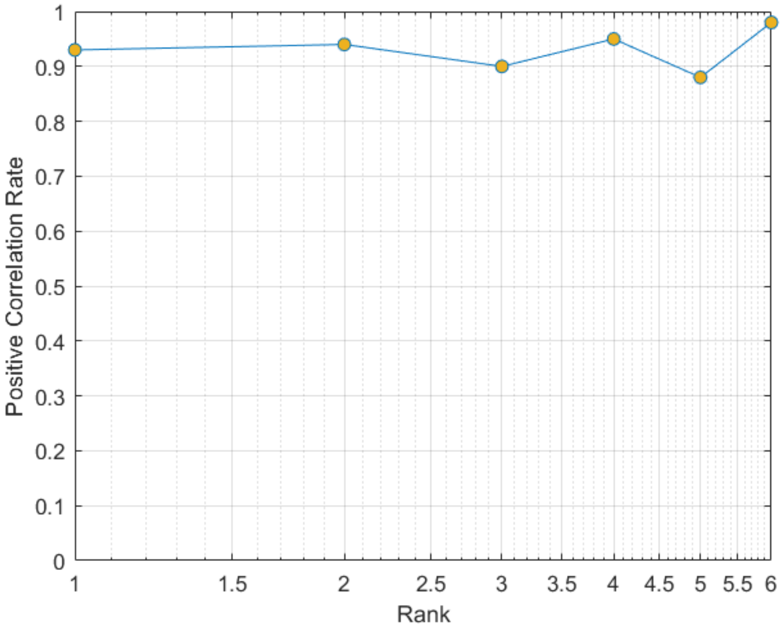


**Figure 4.** Cumulative match characteristics curve of the features from MIC.

### 2.2.4. Features Scaling

*Z*-score transformation is applied to scale independent features at each video frames at some threshold range. Feature scaling is carried out due to learning network employed gradient descent, which converges faster than non-scaled features. Z-score transforms each feature information from zero to its unit variance. Thus, Z-score is given by

$$Z - transform = \frac{(\beta - mean(\beta))}{s.t.d(\beta)} \tag{22}$$

### 2.2.5. Skeletal Video Frames Correction

We use the video frame manipulation (correction) strategy to control initial frame coordinates and position. This is because of the different hand speeds and variations (intuitive interaction) during dynamic word performance. We address this is to highlight the subsequent frame in the sequence, when two or more gestures exhibit different hand trajectories [52]. In what follows, we exploit information of all the frames in the sequence. From each sequence, we calculate the average distance among the frames at $F_P$, $F_Q$ and $F_R$. The average distance is considered for each feature value, which can be utilized to correct the video frames. The technique is mathematically coined as follows:

$$F_P = \frac{\sum_{t=1}^{170}(ValidationSet\beta_{t,Q} - TrainingSet\beta_{t,P})}{170} \tag{23}$$

$$F_Q = \frac{\sum_{t=1}^{170}(ValidationSet\beta_{t,Q} - TrainingSet\beta_{t,Q})}{170} \tag{24}$$

$$F_R = \frac{\sum_{t=1}^{170}(ValidationSet\beta_{t,R} - TrainingSet\beta_{t,R})}{170} \tag{25}$$

where validationSet, TrainingSet, $\beta_t$ and $t$ denote testing information (along $P$, $Q$ and $R$), training information (along P, Q and R), feature vector, and amount of video frames ($t = 1, \cdots, 170$), respectively. This is done by subtracting the first thirty sequences in the feature vector. The three equations make the initial position of each trajectory per frame similar to the frame coordinates. This allows us to compute each dynamic across frames.

## *2.3. ASL Word Recognition from Skeletal Video Features*

The double-hand dynamic ASL word-recognition system is illustrated in Figure 5, which is comprised of the two modules: BiRNN and multi-stacked bidirectional-LSTM.

### 2.3.1. BiRNN Features Extraction

Skeletal joints are automatically extracted using bidirectional recurrent neural network (BiRNN). Empowering the RNN architecture with two BiRNN layers improves the learning behavior with symmetrical, previous and subsequent frame for each information in the video sequence [53] and no re-positioning of the input videos from the ground truth or intended sequence. Nonlinear operations and architecture with hidden layers of BiRNN allow one to find patterns in video sequence. BiRNN is designed and trained using multi-stacked layers in two fashions to extract hand features from skeletal video. We recorded hand gesture video information $v_n$ from input video frame $Q_f$ with sequence length $\Omega$. This input video sequence was fed to BiRNN layer. $v_n$ is defined as $(v \in Q_f)$ where $1 \leq n \leq \Omega$. BiRNN layers received input video sequence $Q_f$, and th function in Equation (26) was evaluated to update its $n$-hidden states, according to the input units $[h_1, h_2, \cdots, Q_t]$, until it learned the last hand gesture video information in the last video frames $v_n = 0$. The information in the present layer is automatically opposite to the hidden units (layers), so the output layer will not update till the hidden units have processed the whole video information. For the backward direction, the total output layer units are computed, and fed back into the hidden layers in opposite directions. The second phase of the BiRNN layers is trained to learn output of the previous layers to be initial state of first layers and yields output vector $\beta_t = [t_1, \cdots, t_\Omega]$, and it is defined as: $t_\Omega \in \beta_t$, where $1 \leq n \leq \Omega$. Finally, BiRNN extraction layers can be written uniformly as [43]:

$$h_t = \sigma[V_{\overrightarrow{hq}}, Q_{f,n} + V_{\overleftarrow{hq}}, Q_{f,n} + V_{\overrightarrow{hh}}, Q_{f-1,n} + V_{\overleftarrow{hh}}, Q_{f-1,n} + d_h]. \quad (26)$$

where dominant and non-dominant hand index is denoted as $n$, and $(\overrightarrow{h_q})$ and $\overleftarrow{h_q}$ denote forward and backward pass hidden state vectors, respectively. In Equation (26), the extraction layers of BiRNN not only give the relationship of video input features vector but also correlate to state of prior sequence.

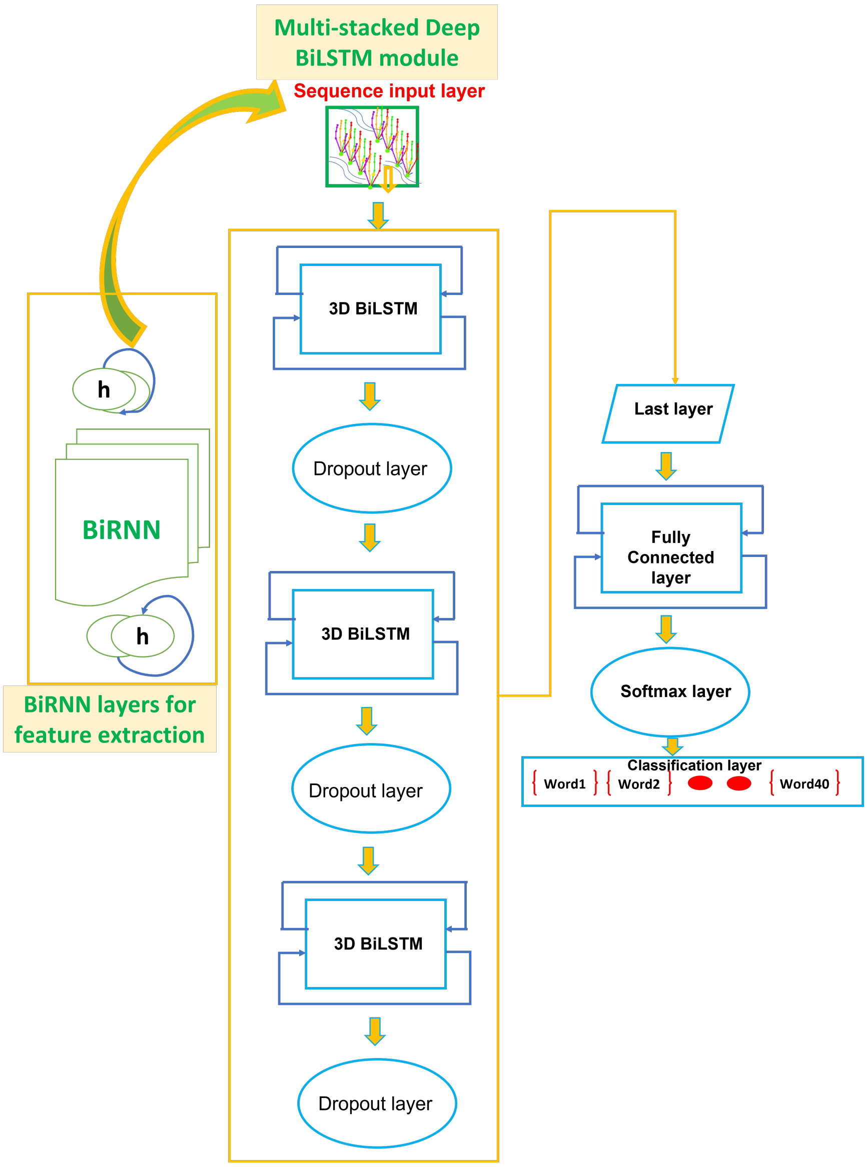


**Figure 5.** Proposed Architecture of Multi-stack Deep BiLSTM.

Moreover, after extracting the matrices of the six selected features, we transformed the matrices into a feature vector. However, many techniques are available for feature transformation such as columns concatenation, rows concatenation and zigzag scheme. As reported in the literature rows, concatenation demonstrates best concatenation. Thus, we convert matrix into feature vector to obtain features in Equation (6). Equation (6) provides training input sequence (six proposed extracted features). The 3D skeletal hand joints are extracted and represented as input features vector, as illustrated in Table 3.

**Table 3.** Extracted features.

| Features | Point of Interest | Description |
|---|---|---|
| Angle points | Pitch, yaw and roll | Hand orientation; 44 skeletal hand joints |
| Relative trajectories | Motion | Hand motion trajectories, frame speed and velocity. |
| Positions | Direction | Arm, palm, wrist and five fingers;<br>(thumb, index, middle, ringy and pinky) |
| Finger joints | Coordinates | Coordinate of five fingers' tip, metacarpal,<br>proximal, distal and interdistal. |
| Hand pause | Motion | Hand preparation and retraction. |

### 2.3.2. Long-Short Term Memory (LSTM)

LSTM is a family of RNN to handle gradient vanishing, by substituting an extended bidirectional LSTM (BiLSTM) neurons [27,54,55]. BiLSTM neuron learn long-term dependencies between sequences [5,31,56]. Single BiLSTM unit return low accuracy especially when learning complex sequences. Deep BiLSTM is introduced to enhance accuracy of single LSTM unit. Multiple long short-term memory (known as deep BiLSTM) architecture is the strategy of concatenating number of BiLSTM hidden units in fashionable manner. This is to achieve high-level sequential representations from sequential video information. In deep BiLSTM, output of former layer $l$-1 serves as sequence input to present layer $l$. Results demonstrated that deeper networks improve recognition performance [36].

Deep BiLSTM network is illustrated in Figure 5, which is realized by concatenating three-additional BiLSTM layers with output mode "sequence" before each BiLSTM layer. Dropout layer is connected after each BiLSTM layer to control overfitting and alter fundamental network architecture, which is defined in Equation (27) [57]. The final output of all sequences is concatenated to construct one output layer known as softmax layer. The output mode of last BiLSTM layer is now coined as "last". Therefore, ASL words class prediction is conducted by equipping the last layer of BiLSTM network with classification layer. Classification layer is configured with cross entropy loss function [58]. The fully connected layer multiplies sequential input by weight $\alpha$ and then adds $\rho$. However, fully connected layer merges all features in $\beta_t$ to classify word gesture. In our case, information from fully connected layer of deep multi-stack BiLSTM network is exactly the same as the number of word classes of sequential features. This procedure is known as multi-stacking, and the architecture is referred to as multi-stack deep BiLSTM.

$$rand(size(d_i) < probability \tag{27}$$

where $d_i$ denotes drop layer input.

### 2.3.3. Multi-Stacked Deep BiLSTM Training from Transfer Learning

The major limitation of training multi-stacked deep BiLSTM network is the high demand for large input video set. The number of our input video sets is moderate. However, training a new BiLSTM network is a complex and costly process. Multi-BiLSTM network from the existing method has large number of abstractions, and this makes learning difficult. This can lead to misclassification. To overcome this problem, transfer learning (TL) via deep neural network is extended to SLR. TL is a methodology of utilizing a pretrained deep network that has proven successful as initial step (newly designed network) to learn feature from unknown signer. A methodology of fine-tuning network brings simple and fast learning network, compared to conventional network initialized from the grass-root. Researchers identified the potential of neural-network-based TL [59,60]. In this paper, TL approach based on multi-stack deep BiLSTM network as shown in Figure 6 is implemented to recognize dynamic double-hand ASL words. Extracted input feature vector in Equation (6) is built into multi-stacked BiLSTM layers for double-hand dynamic ASL words recognition. Multi-stacked BiLSTM is trained to obtain output probability vectors for all of its corresponding input vectors, predicted word classes, and confusion matrices. Multi-stacked layers are initialized with weight of extracted features, as follows [43]:

$$\begin{aligned}O_{t,n} = [&V_{\overrightarrow{h}o}\overrightarrow{h}_{f}, P^{f}_{t,n} + V_{\overleftarrow{h}o}\overleftarrow{h}_{f}, P^{f}_{t,n} + V_{\overrightarrow{h}o}\overrightarrow{h}_{f}, \omega^{f}_{t,n} + \\ &+ V_{\overleftarrow{h}o}\overleftarrow{h}_{f}, \omega^{f}_{t,n} + V_{\overrightarrow{h}o}\overrightarrow{h}_{f}, \eta_{t,n} + \\ &+ V_{\overleftarrow{h}o}\overleftarrow{h}_{f}, \eta_{t,n} + V_{\overrightarrow{h}o}\overrightarrow{h}_{f}, N_{t,n} + V_{\overleftarrow{h}o}\overleftarrow{h}_{f}, N_{t,n} + \\ &+ V_{\overrightarrow{h}o}\overrightarrow{h}_{f}, K_{t,n} + V_{\overleftarrow{h}o}\overleftarrow{h}_{f}, K_{t,n} + \\ &+ V_{\overrightarrow{h}o}\overrightarrow{h}_{f}, \varphi_{t,n} + \\ &+ V_{\overleftarrow{h}o}\overleftarrow{h}_{f}, \varphi_{t,n} + d_{o}].\end{aligned} \tag{28}$$

The final classification layer is formulated as follows:

$$O = \sum_{\Omega=0}^{\Omega-1} O^{t}_{\Omega,n} \tag{29}$$

$$O^{t} = p(E_{L}|\beta) = \frac{e^{O^{L}}}{\sum_{i=0}^{L-1} e^{O_{i}}}, L = 1, \cdots, L \tag{30}$$

where $L$ and $O^{t}_{\Omega}$ denote ASL word classes and predicted probability class $E_{L}$ when ASL word features $\beta$ is given, respectively. However, softmax function transforms the output value into [0, 1] and transforms the weight of $L$ values into 1. The ground truth is given as $O^{L} \in [0,1]$, as well as prediction probabilities as $\overrightarrow{O^{L}}$. The network parameters can be given as in Equation (31), as follows:

$$\theta[u+1] = \theta[u] + r(-\frac{\partial O}{\partial \theta[u]}), \theta[0] \approx U[0,1]. \tag{31}$$

From Equation (31), $\theta[u]$, $u$ and $r$ denote parameters set, parameter update times, and learning rate. This equation consists of all weights and biases in the Deep BiLSTM network.

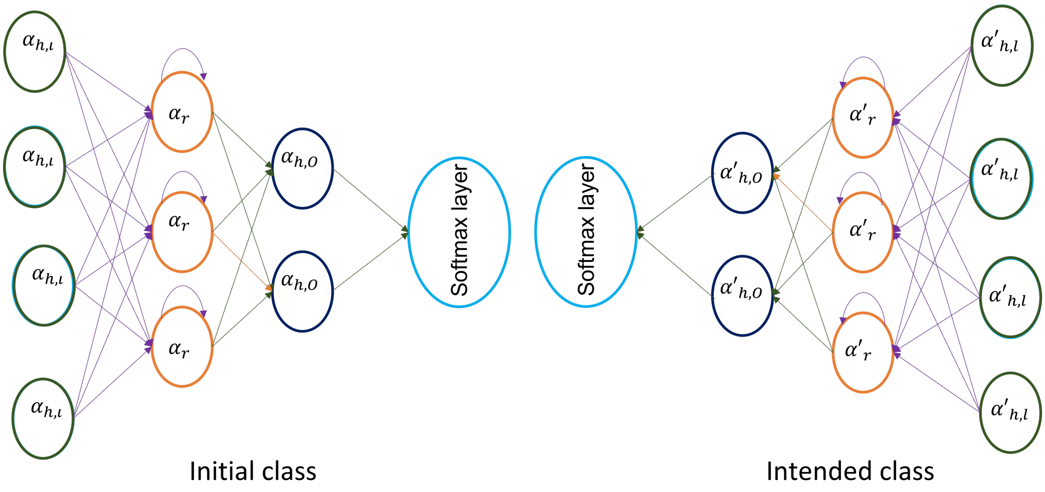


**Figure 6.** Diagram of transfer deep LSTM network.

Let an initial class $C_{i} = \beta_{i}$ have a learning period $p_{i}$; thus, the intended class $C_{d} = \beta_{d}$ has a learning period $p_{d}$. Thus, the aim is to aid learning the prediction function of the intended class by utilizing knowledge gained by $p_{i}$ from initial class. However, transfer learning has a rule: that the initial class should be different from the intended class, as well as their learning periods. For the intended class, we have recorded 40 ASL words from 10 signers, which are repeated 10 times, making a total of 4000 samples, whereas for the initial class, we have recorded 10 signers different from the ones in the intended class; however, each signer performs 58 ASL words, 10 times, which makes a total of 58,000 sequences. For details of the experimental set up and data recording process, see Section 2.6.1.

The feature learning phase of the network has five layers, as illustrated in Figure 5. In Figure 6, the features of a successful network can be reused in a newly adopted network. The weight matrices among input and the hidden layers $\alpha_{h,I}$, recurrent weight matrices in the hidden layers $\alpha_r$, and connection weight matrices among hidden layers and output layer $\alpha_{h,o}$ were trained in the initial class (trained in advance with sufficient features). The successful network is illustrated in Figure 5. Thus, the weight matrices among input layers and hidden layers are transferred to intended class features as weight initial value. This new approach of weight initialization is superior to random initialization. However, training features of intended class were used to adjust the BiLSTM weights on small data set. Thus, recurrent weight matrices in the hidden layers, and connection weight matrices among hidden layers and output layer, were initialized at random.

### 2.4. Model Parameters

The selected method is experimentally validated with careful selection of parameters in Table 4. These parameters were achieved through cross-validation. Our experiments are designed from personal computer (PC) on Windows 10 operating system equipped with CPU Core i7 9th Gen, 8 GB RAM, details of the execution environment is provided in Table 5. Serial communication from LMC to PC is enabled via written C# codes on Microsoft visual studio environment, and LabView library.

**Table 4.** Network parameter selection.

| Network Layers | Parameter Options | Selection |
|---|---|---|
| Input layer | Sequence length | Longest |
| | Batch size | 27 |
| | Input per sequence | 170 |
| | Feature vector | 1 dimension |
| Hidden layer | 3 Bi-LSTM layers | Longest |
| | Hidden state | 200 |
| | Activation function | Softmax |
| Output layer | LSTM model | Many to one |
| | Number of classes | 40 |

**Table 5.** Execution environment.

| Deployment | Descriptions |
|---|---|
| PC | Dell<br>CPU: Intel Core i7-9th Gen<br>Memory Size: 8 GB DDR4<br>SSD: 500 GB |
| LMC sensor | Frame rate: 64 fps<br>Weight: 32 g<br>Infrared camera: 2 × 640 × 240<br>Range: 80 cm<br>150 × 120° |
| Participants | Ten |
| Selections | 40 ASL words<br>10 times number of occurrence |

### 2.5. Evaluation Metrics

Confusion matrix contains columns and rows, where each column denotes possibility of predicted word gestures, whereas each row denotes original word gesture probability.

However, main diagonal of confusion matrix denotes scores of correct classified word gestures with blue colors, whereas entries below diagonal denote false positives (gestures classified incorrect from our concerned class) with gold color, and entries above diagonal denote false negatives (gestures classified incorrect from non-concerned class) with dark orange color. From confusion matrix, set of word pairs inside similar cell and in similar class is denoted as true positive,$\tau_1$; set of word pairs inside similar cell and in different class is denoted as true negative, $\tau_2$; set of word pairs inside different cell and in different class is denoted false positive, $\psi_1$; and set of word pairs in different cells and in different classes is denoted false negative, $\psi_2$. Each word pair is computed based on its frequency of occurrences. However, it is demonstrated that $\tau_1$ and $\psi_2$ should be maximized and $\tau_2$ and $\psi_1$ minimized to better explore performance of selected features and to determine optimal multi-stacked deep BiLSTM recognition. The following metrics are most popular for deep neural network and provide the results of comparison [5].

### 2.5.1. Accuracy Metrics

*Accuracy* is described as measure of correct predictions. *Accuracy* is given by:

$$Accuracy = \frac{\tau_1 + \tau_2}{\tau_1 + \tau_2 + \psi_1 + \psi_2} \tag{32}$$

Furthermore, accuracy index is not resourceful when two word classes are of varied sizes; this leads one to obtain high measure of correct predictions. To overcome this daunting problem, the following indices are augmented as best choices [61]:

### 2.5.2. Fowlkes–Mallows (*FI*) Index

Fowlkes–Mallows index is adopted to evaluate level of similarity between trained and predicted word classes.

$$FI = \sqrt{\frac{\tau_1}{\tau_1 + \tau_2} * \frac{\tau_1}{\tau_1 + \psi_1}} \tag{33}$$

### 2.5.3. Matthew Correlation Coefficient (*MC*)

Matthew correlation coefficient determines trained and predicted binary classification [62], which is defined as:

$$MC = \frac{(\tau_1 * \tau_2) - (\psi_1 * \psi_2)}{\sqrt{(\tau_1 + \psi_1)(\tau_1 + \psi_2)(\tau_2 + \psi_1)(\tau_2 + \psi_2)}} \tag{34}$$

### 2.5.4. Sensitivity $(S_v)$

Sensitivity is defined as:

$$S_v = \frac{\tau_1}{(\tau_1 + \psi_2)} \tag{35}$$

### 2.5.5. Specificity $(S_f)$

Specificity is defined as:

$$S_f = \frac{\tau_2}{(\tau_2 + \psi_1)} \tag{36}$$

### 2.5.6. Bookmaker Informedness (*BI*)

Bookmaker informedness determines probability estimate of informed decision; it is defined as:

$$BI = (S_v + S_f) - 1 \tag{37}$$

### 2.5.7. Jaccard Similarity Index (*JI*)

*JI* metrics describes portion of overlap between two words: word 1 (trained word) and word 2 (generalized word), where they share similar features. These features are considered 0 or 1. Each feature per particular class must fall into one of $\tau_1$, $\psi_2$, $\tau_2$ and $\psi_1$ entries, respectively. *JI* is given as [63]:

$$JI = \frac{\tau_1}{(\tau_1 + \tau_2 + \psi_1)} \tag{38}$$

Moreover, the developed models from the proposed method are evaluated using the accuracy, sensitivity, and specificity metrics. However, the best model is subject to further evaluation using K-fold and LOSO cross-validation to observe the influence of majority over the minority classes (class imbalance). The shortcomings of these recognition metrics include displaying misguiding results on imbalanced features due to failure to accommodate the relationship between the positive and negative entries in the confusion matrix. In addition, these metrics were not good enough to evaluate the matrix overlap. Therefore, to monitor the exact classification accuracy of our best model and to overcome the limitations of the mentioned metrics, we extend the evaluation of *MC*, *JI*, *BI* and *FI* indices. These metrics were reported in some studies to demonstrate good performance.

### *2.6. Experiment*

In this section, we present the experimental procedures of the implemented system. The system is implemented using best hardware selection details on Table 5, which are assembled to provide the experimental set up of Figure 7. In the simulation task, several Matlab packages were used to validates the network performance.

#### 2.6.1. Dataset Design

Available public hand skeletal ASL datasets with resourceful 3D skeletal hand information while signer is on the move, as in our proposal, are scanty, thus making it necessary to construct our data sets. In this approach, we selected 40 dynamic double-hand ASL words from first 200 available ASL words vocabulary. All signs were captured from 10 right-handed (right hand as dominant hand) double-hand signers. We extended strategy for in-the-field data design in [27]. All signers were trained from web ASL video information tutors. Age range of signers was 25–40 years . Each signer repeated double-hand ASL word 10 times, making a total of 4000 samples. LMC is suspended on signer's chest, as shown in Figure 7, to actualize ubiquitous sign language recognition system. LMC is a vision-based capturing devices that employs infrared image sensing analogy at 30 frame per second, with 2 × 640 × 240 range to extract 3D hand-joint skeletal video information. LMC SDK software is configured via API (application programming interface) to synchronize with MS visual studio and LabView frameworks for data recording and visualization. Brief description of our designed data set is details in Table 6. We recorded 170 frames per each 131 skeletal video sequence length. However, some video frames contained sequence length less than 131. Then, we applied padding procedures in [32] to obtain equal number of sequence length. Our adopted network was further validated on the three challenging public-hand skeletal dynamic gestures from LMC data bases as follows. These data sets were evaluated according to the leave-one-subject-out experimental protocol:

- Avola et al. [32] data set: the data set is comprised of static and dynamic skeletal hand gestures captured from 20 signers, and it is repeated twice. Due to the nature of our approach, we selected dynamic gestures, including bathroom, blue, finish, green, hungry, milk, past, pig, store and where.
- LMDHG [64] data set: comprised of dynamic skeletal hand gestures collected from 21 signers, each signer performed at least one sign, resulting in 13 ± 1 words.
- Shape Retrieval Contest (SHREC) [65] data set: Comprised of 14 and 28 challenging dynamic skeletal hand gestures, respectively. The gestures were performed using one finger and the entire hand.

**Table 6.** Dataset description.

| Classes | Amount |
|---|---|
| Frames | 524,000 |
| Samples | 4000 |
| Duration (sec) | 8200 |

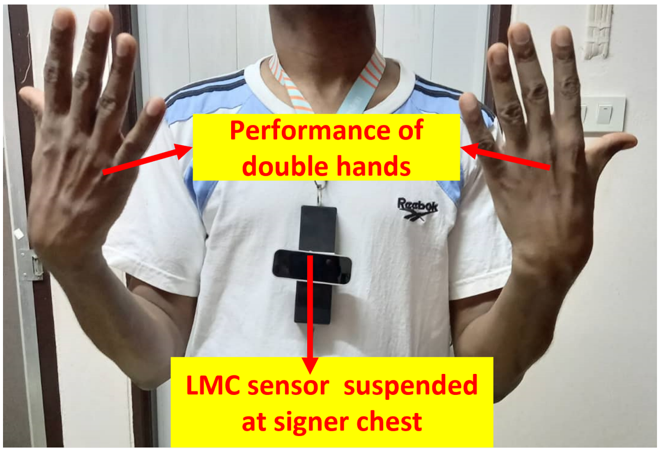

Performance of double hands
LMC sensor suspended at signer chest

**Figure 7.** Experimental set up.

## 3. Results

In this section, we present simulation results of the adopted multi-stacked deep BiLSTM networks. Two type of deep networks were designed and simulated to demonstrate accuracy of our selected features, as shown in Table 7.

**Table 7.** Proposed models combination.

| Models | | Epochs | Execution Time (s) |
|---|---|---|---|
| Features Combination | Model Depth | | |
| Shape + Motion + Position + Angles + Pause + Relative trajectory | 3-BiLSTM layers | 300 | 1.05 |
| Shape + Motion + Position + Angles | 3-BiLSTM layers | 300 | 1.01 |

The first network combined hand shape, motion, position, pause, angle and relative trajectory. After several trial and error parameter selections, it was found that best model for different input feature combinations settled at Model-1 with 3 multi-stacked deep BiLSTM layers. Model-1 was trained at 300 epochs, where each class pair had inferences at 1.05 s, as illustrated in Figure 8 and Table 7. The second network was made through combination of shape, motion, position, and angle features. After several trials of network training for different feature combinations, best model was settled at Model-2 with 3 multi-stacked deep BiLSTM units, inferences at 1.01 s via 300 epochs, as illustrated in Figure 9 and Table 7. Since Model-1 achieved best recognition, we subjected it to further analysis using Leave-One-Subject-Out (LOSO) protocol because of its robustness, where 9 signers out of 10 were trained and the remaining signer was used during validation (generalization). This procedure was repeated 10 times, and the results are reported in Table 8. We achieved best LOSO validation due to reduced cost from the TL. Good discrimination performance was noticed by the developed multi-stacked deep BiLSTM when knowledge of hand dynamics were used in the input vector, achieving average sensitivity of 97.494%, specificity of 96.765%, average *FI* of 72.347%, *MC* of 94.937%, *BI* of 94.259%, *JI* of 54.476% and accuracy rate up to 97.983%. Therefore, the two models were set to inference with Top-K validation [66]. The data set was partitioned into 80% and 20% for training and validation, respectively. In this trial, K took values of 1, 2 and 3. Results of Top-3 validation are demonstrated in Table 9. It is demonstrated that model-1 achieved best accuracy of the three classes ($k$ = 1, 2, 3). This indicates that additional feature from pause and relative trajectory (knowledge of hand dynamics with motion speed) contributed to 4% accuracy when compared to second model with only four input features. Table 10 summarizes the computing cost required to test our best model. An ablation investigation of our designed data set revealed the influence of stacking multiple BiLSTM layers. The multi-stacked BiLSTM network was trained using the three network performance schemes to optimize the loss function. Figures 10–12 demonstrate the recognition performance of multi-stacked deep BiLSTM network with optimization from Adam, stochastic gradient descent and adaptive gradient schemes, respectively. Their performance comparison of computed mean of standard evaluation metrics is displayed in Table 11, which is obtained by condensing the entire confusion matrix for the average results. The best optimization scheme for multi-stacked deep BiLSTM with Model-1 input feature vector is Adam.

Table 12 provides performance comparison between average recognition accuracy of Model-1 and proposed method in [32]. The work [32] has similar shape with our approach because this method utilized gestures from ASL dictionary. Their method employed 20 signers, and each signer was directed to perform 12 dynamic double- and single-hand ASL words, 30 times each. Our multi-stack deep BiLSTM network was outperformed [32] on ASL data set with accuracy, precision, recall and F1-Score of 1.48%, 1.597%, 1.469%, and 1.626%, respectively. These results are consistent with the skeletal dynamic hand gesture recognition. When our method was validated on LMDHG data set, it was outperformed [32] with mild recognition accuracy of 0.37%. In addition, our method was validated on SHREC data set, and work in [32] was superior to our technique by 0.63% for experiment with 14 hand gestures, while we outperformed [32] by 1.56% for experiment with 28 hand gestures.

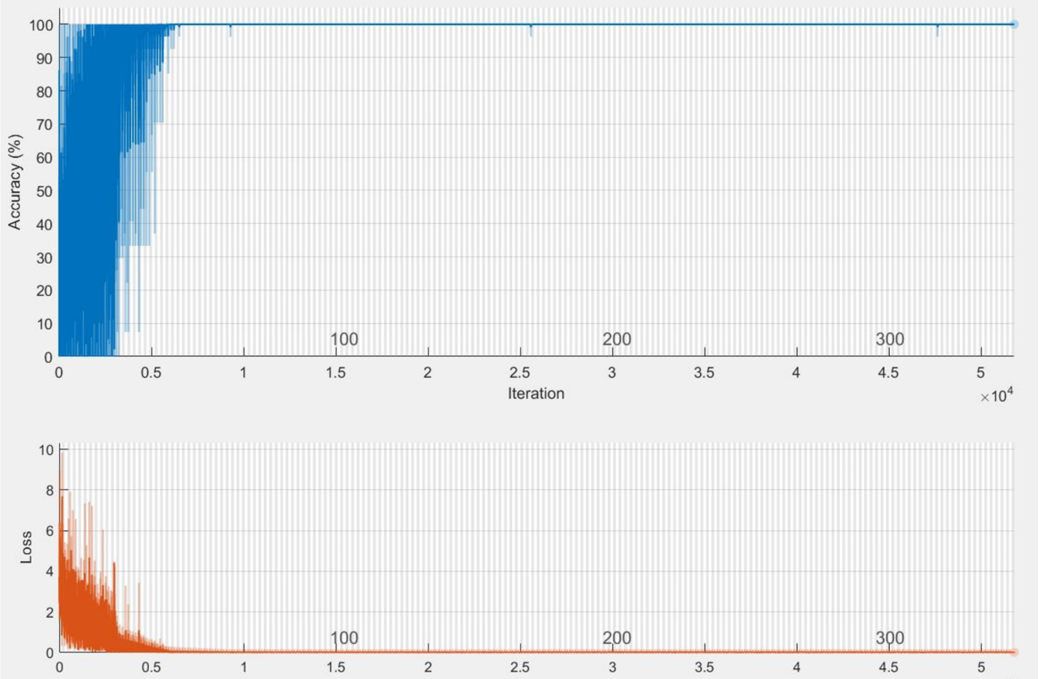


**Figure 8.** Training performance of Deep BiLSTM network on Model-1.

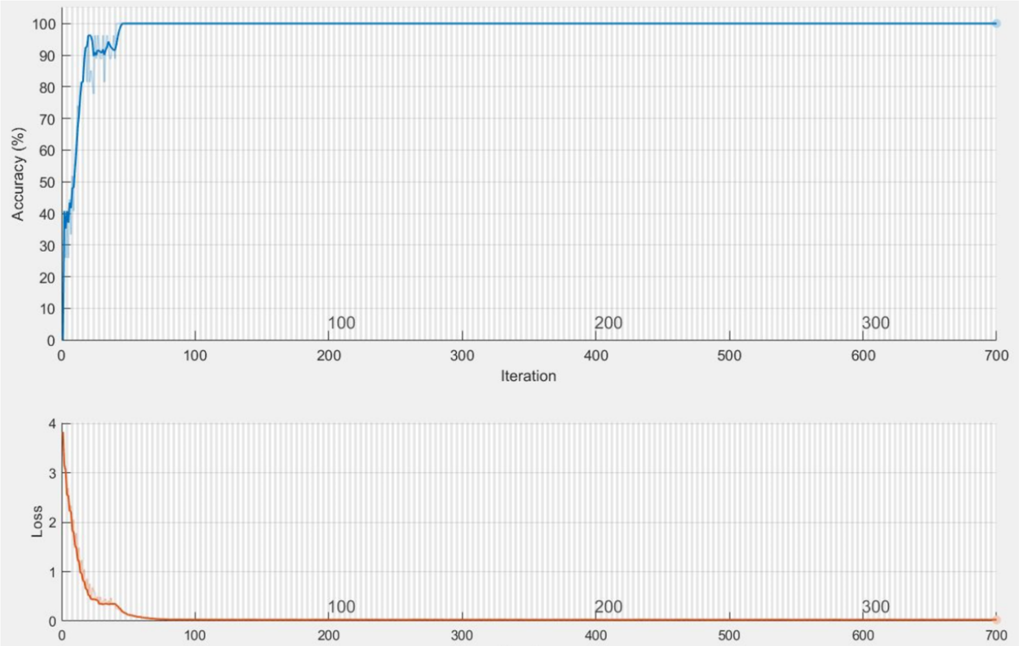


**Figure 9.** Training performance of Deep BiLSTM network on Model-2.

**Table 8.** Performance evaluation of multi-stack deep BiLSTM network using leave-one-subject-out cross-validation.

| Word Classes | *Accuracy* | *FI* | *MC* | $S_v$ | $S_f$ | *BI* | *JI* |
|---|---|---|---|---|---|---|---|
| Again | 0.98 | 0.7 | 1 | 0.98 | 0.992361 | 0.972361 | 0.494949 |
| Angry | 0.92 | 0.707721 | 0.842701 | 0.888889 | 0.956522 | 0.845411 | 0.510638 |
| Available | 0.986486 | 0.582301 | 0.970193 | 0.970874 | 0.994819 | 0.965692 | 0.341297 |
| Bad | 0.99 | 0.703562 | 0.959462 | 0.99 | 0.985622 | 0.975622 | 0.497487 |
| Bicycle | 0.993197 | 0.579324 | 0.984886 | 0.98 | 0.972569 | 0.952569 | 0.335616 |
| Big | 0.984772 | 0.707251 | 0.969581 | 0.99 | 0.979381 | 0.969381 | 0.505102 |
| But | 0.980198 | 0.989899 | 0.489899 | 0.989899 | 0.5 | 0.489899 | 0.98 |
| Car | 1 | 1 | 0.899994 | 1 | 0.989457 | 0.989457 | 1 |
| Cheap | 0.975 | 0.701793 | 0.943489 | 0.970297 | 0.999673 | 0.96997 | 0.497462 |
| Clothes | 0.97 | 0.703599 | 1 | 0.960784 | 0.986117 | 0.946902 | 0.5 |
| Cold | 0.994898 | 0.716115 | 0.989841 | 0.990099 | 1 | 0.990099 | 0.512821 |
| Come | 1 | 1 | 0.959284 | 1 | 0.98884 | 0.98884 | 1 |
| Dance | 0.975 | 0.701793 | 0.963497 | 0.970297 | 0.989916 | 0.960213 | 0.497462 |
| Embarrassed | 0.955 | 0.709103 | 0.976592 | 0.933333 | 0.979465 | 0.912798 | 0.507772 |
| Empty | 0.98 | 0.7 | 0.965484 | 0.98 | 0.969476 | 0.949476 | 0.494949 |
| Excuse | 0.98 | 0.7 | 1 | 0.98 | 0.981238 | 0.961238 | 0.494949 |
| Expensive | 0.98 | 0.7 | 0.974596 | 0.98 | 0.961296 | 0.941296 | 0.494949 |
| Finish | 1 | 1 | 0.958249 | 1 | 0.952948 | 0.952948 | 1 |
| Fork | 0.975 | 0.701793 | 0.976222 | 0.970297 | 0.983176 | 0.953473 | 0.497462 |
| Friendly | 0.993103 | 0.571305 | 0.984467 | 0.979167 | 1 | 0.979167 | 0.326389 |
| Funeral | 0.985 | 0.698221 | 0.965785 | 0.989899 | 0.976355 | 0.966254 | 0.492462 |
| Go | 1 | 1 | 1 | 1 | 0.954389 | 0.954389 | 1 |
| Good | 1 | 1 | 0.969829 | 1 | 0.965481 | 0.965481 | 1 |
| Happy | 0.975 | 0.701793 | 0.949985 | 0.970297 | 0.968367 | 0.938664 | 0.497462 |
| Help | 0.98 | 0.7 | 0.947892 | 0.98 | 0.983923 | 0.963923 | 0.494949 |
| Jump | 0.975 | 0.701793 | 0.928965 | 0.970297 | 0.976583 | 0.94688 | 0.497462 |
| March | 0.994898 | 0.716115 | 0.989841 | 0.990099 | 1 | 0.990099 | 0.512821 |
| Money | 0.975 | 0.701793 | 0.939785 | 0.970297 | 0.954873 | 0.92517 | 0.497462 |
| Please | 0.912458 | 0.485965 | 0.801818 | 0.930233 | 0.905213 | 0.835446 | 0.274914 |
| Pray | 0.984694 | 0.705419 | 0.969432 | 0.989899 | 0.979381 | 0.96928 | 0.502564 |
| Read | 0.98 | 0.7 | 0.985672 | 0.98 | 0.972345 | 0.952345 | 0.494949 |
| Request | 0.98 | 0.7 | 0.925498 | 0.98 | 0.991438 | 0.971438 | 0.494949 |
| Sad | 0.979899 | 0.712525 | 0.960582 | 0.961165 | 1 | 0.961165 | 0.507692 |
| Small | 0.989712 | 0.444416 | 0.968427 | 0.95 | 1 | 0.95 | 0.197505 |
| Soup | 0.935 | 0.694709 | 0.977349 | 0.92233 | 0.965481 | 0.887811 | 0.494792 |
| Spoon | 0.979452 | 0.573573 | 0.954375 | 0.97 | 0.984375 | 0.954375 | 0.33564 |
| Time | 0.98 | 0.7 | 0.982396 | 0.98 | 0.974928 | 0.954928 | 0.494949 |
| Want | 0.994819 | 0.714435 | 0.989686 | 0.989899 | 1 | 0.989899 | 0.510417 |
| What | 0.994819 | 0.714435 | 0.989686 | 0.989899 | 1 | 0.989899 | 0.510417 |
| With | 0.984694 | 0.697926 | 0.969432 | 0.979381 | 0.989899 | 0.96928 | 0.489691 |
| AVERAGE | 0.979827 | 0.723467 | 0.949372 | 0.974941 | 0.967648 | 0.942588 | 0.54476 |

**Table 9.** Performance accuracy of the developed models.

| Network Models | Top-1 | Top-2 | Top-3 | Features Combination | Model Depth |
|---|---|---|---|---|---|
| Deep BiLSTM | 0.954 | 0.971 | 0.989 | Shape + Motion + Position + Angles + Pause + Relative trajectory | 3-BiLSTM layers |
| Deep BiLSTM | 0.912 | 0.929 | 0.945 | Shape + Motion + Position + Angles | 3-BiLSTM layers |

**Table 10.** Comparison of adopted Deep Bi-LSTM with a state-of-the-art method.

| Methods | Type of Deep Learning | No. of Epochs | Depth of LSTM | Convergence Rate | Execution Time |
|---|---|---|---|---|---|
| Avola et al. [32] | Deep Bi-LSTM | 800 | 4 units | 100,000 iter | not reported |
| Our proposal | Deep Bi-LSTM | 300 | 3 units | 10,000 iter | GPU 1002 |

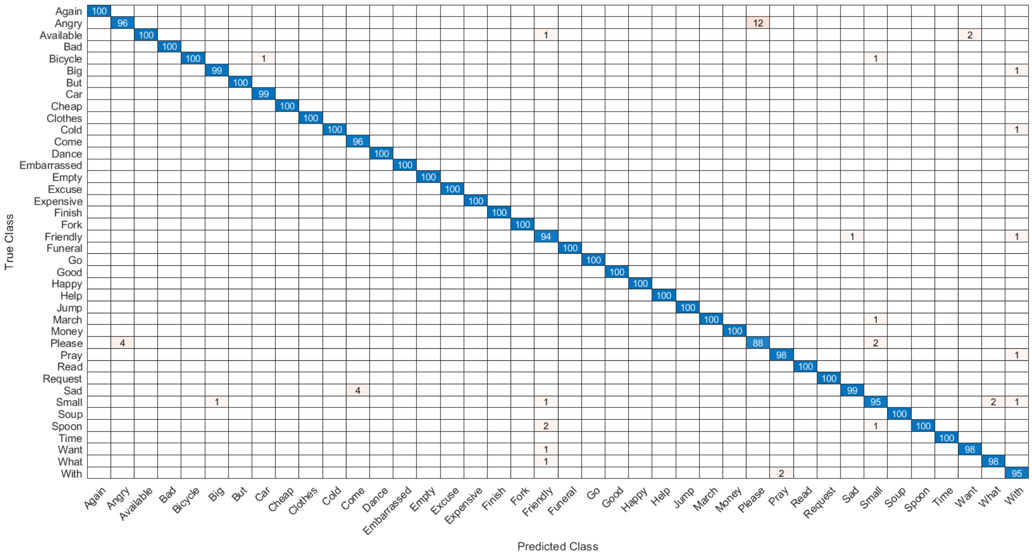


**Figure 10.** Confusion matrix of the recognition performance of double-hand dynamic ASL words with Adam optimization.

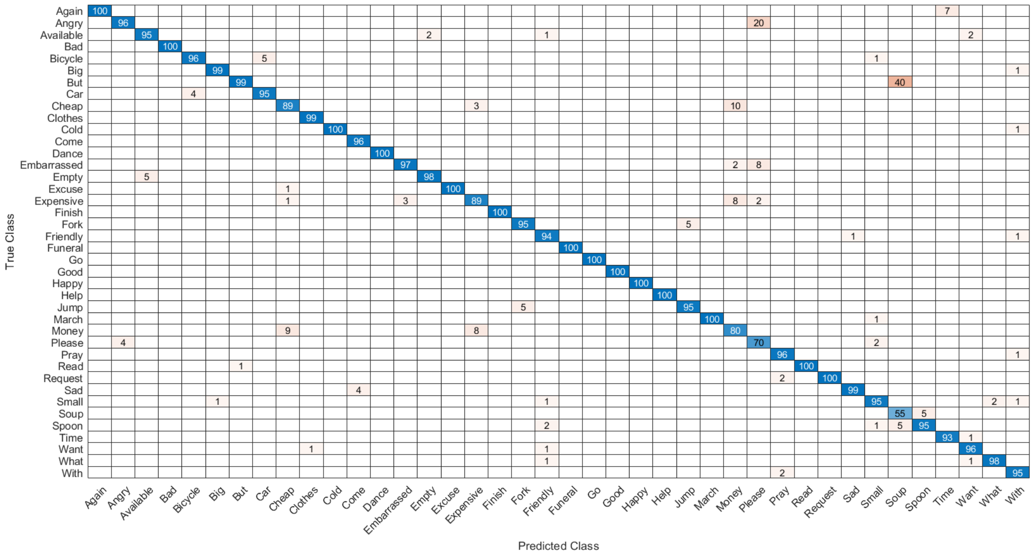


**Figure 11.** Confusion matrix of the recognition performance of double-hand dynamic ASL words with SGD optimization.

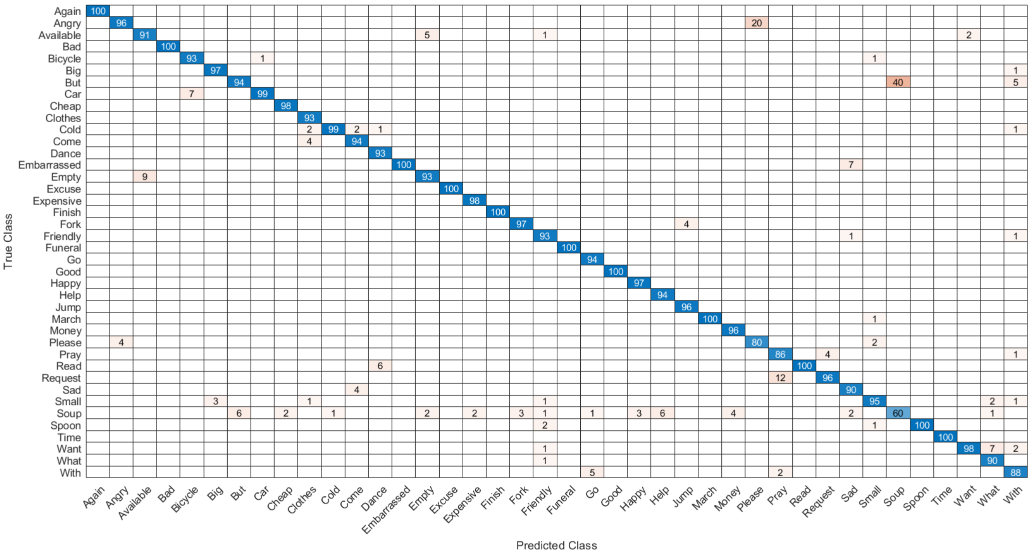


**Figure 12.** Confusion matrix of the recognition performance of double-hand dynamic ASL words with Adagrad optimization.

**Table 11.** Performance validation of Multi-stack deep BiLSTM from adopted optimization scheme.

| Optimization Scheme | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| AdaGrad | 94.701 | 94.006 | 94.869 | 94.003 |
| SGD | 95.011 | 94.998 | 95.01 | 94.786 |
| Adam | 97.983 | 96.765 | 97.494 | 96.968 |

**Table 12.** Performance comparison of the multi-stacked BiLSTM network with method in [32].

| | **ASL Data Set** | | | |
|---|---|---|---|---|
| Approach | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
| Avola et al. [32] | 96.4102 | 96.6434 | 96.4102 | 96.3717 |
| Ours | 97.881 | 98.007 | 97.879 | 97.998 |
| | **LMDHG Data Set** | | | |
| Avola et al. [32] | 97.62 | | | |
| Ours | 97.99 | | | |
| | **SHREC Data Set** | | | |
| | Accuracy (%) | | | |
| | 14 Hand Gestures | 28 Hand Gestures | | |
| Avola et al. [32] | 97.62 | 91.43 | | |
| Ours | 96.99 | 92.99 | | |

*Performance Comparison with Baseline Methods*

Validation is carried out with various baselines on the LMDHG and SHREC'17 databases, respectively. Different results are illustrated and analyzed.

In Table 13, evaluation results of SHREC'17 dataset from standard protocol are illustrated. Methods Avola et al. [32] and Li et al. [67] are in similar shape with our approach, and their results are obtained from [65,68]. In particular, our method obtains 96.99% on the 14-gesture protocol and 92.99% on the 28-gesture protocol. It outperforms the most recent works [67,68] by 0.48% and 0.68% for experiment with 14 hand gestures and by 1.5% in [68] for experiments with 28 hand gestures, respectively, though [67] is superior to our technique by 0.34% for experiment with 28 gestures. However, our method demonstrates state-of-the-art performance on recent approaches. Table 14 illustrates evaluation results of LMDHG data set. The comparison results are obtained from works in [32,64]. Our method outperforms the two recent approaches in [26,35] on LMDHG data set with average recognition accuracies of 6.79% and 5.99%.

**Table 13.** Performance of the multi-stacked BiLSTM network initialized with data-driven optimization applied to SHREC data set.

| Approach | Algorithms | Accuracy (%) | |
|---|---|---|---|
| | | 14 Hand Gestures | 28 Hand Gestures |
| De Smedt et al. [65] | SVM | 88.62 | 81.9 |
| Boulahia et al [69] | SVM | 90.5 | 80.5 |
| Ohn-Bar and Trivedi [70] | SVM | 83.85 | 76.53 |
| HON4D [71] | SVM | 78.53 | 74.03 |
| Devanne et al. [72] | KNN | 79.61 | 62 |
| Hou et al. [73] | Attention-Res-CNN | 93.6 | 90.7 |
| MFA-Net [74] | MFA-Net, LSTM | 91.31 | 86.55 |
| Caputo et al. [75] | NN | 89.52 | - |
| DeepGRU [76] | DeepGRU | 94.5 | 91.4 |
| Liu et al. [68] | CNN | 94.88 | 92.26 |
| Li et al. [67] | 2D-CNN | 96.31 | 93.33 |
| Ours | Multi-stack deep BiLSTM | 96.99 | 92.99 |

**Table 14.** Performance of the multi-stacked BiLSTM network initialized with data-driven optimization applied to LMDHG data set.

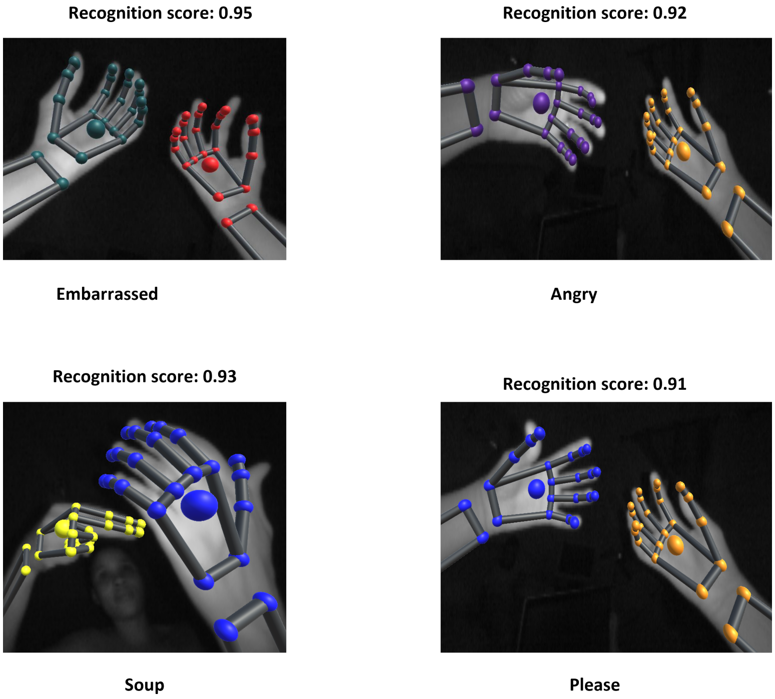
| Approach | Algorithms | Accuracy (%) |
|---|---|---|
| Boulahia et al. [64] | SVM | 84.78 |
| Devanne et al. [72] | KNN | 79.61 |
| Lupinetti et al. [35] | CNN-ResNet50 | 92 |
| Hisham and Hamouda [26] | Ada-boosting | 91.2 |
| Ours | Multi-stack deep BiLSTM | 97.99 |

## 4. Discussion

We combined two set of models from two different input feature set combination to improve hand feature recognition and examine sensitivity per features against recognition accuracy. Sometimes, true and false negatives revealed zero values (with best true positive), and evaluating these values using standard metrics produced a misleading conclusion. Therefore, for better explanation of confusion matrix, true positives and false negatives should be maximized, whereas true negatives and false positives should be minimized, so that sensitivity of adopted algorithms will be effective on the tested features. Accuracy is not enough to describe performance of model-1; however, we evaluated model-1 according to other metrics. We address this problem using the evaluation metrics in Equations (33), (34), (37) and (38), respectively. Figure 10 displays confusion matrix of model-1, which illustrated that true positives and false negatives were the largest entries in the matrix, whereas true negatives and false positives were the lowest entries, respectively. Nevertheless, to conform that the results are statistically significant, the $JI$ is computed using Equation (38) by counting the number of accuracy of similar classes $\geq$ 54.476%. Simulation results demonstrated that $JI$ is up to 0.5. Thus, the similarity index was rejected, leading to the conclusion that the adopted system was statistically significant. Moreover, in order to assess the imbalanced samples (overoptimistic estimation of the classifier ability on the majority class to be dominant) of the adopted multi-stacked deep BiLSTM network, we evaluated MC index from Equation (34). MC generated a high score only if the multi-stacked BiLSTM recognizer was able to correctly predict the majority of positive feature instances and the majority of negative feature instances. MC ranges in the interval with extreme values {–1 and +1} were obtained in case of perfect misclassification and classification, respectively. The MC in this case achieved an average score of 0.949. MC computed results show that the adopted network was able to successfully classify the new input features without minority or majority class bias, reporting only four false negatives (But, Angry, Car and Please), whereas four ASL words (Again, Clothes, Excuse and Go) in the feature vector were all correctly classified (for $\varphi_1$ = 0 or $\varphi_2$ = 0), in this case, MC = 1. In FI computation, if each class in training feature perfectly matches with class in testing feature, then FI is 1, while if each class in training feature is equally shared over the entire classes in testing feature, then FI is 0. Therefore, FI index achieved good matrix overlap of 0.723 in Table 8. Furthermore, model-1 was evaluated using BI Formula (37), where the average gauge of the likelihood of the informed decision reveald a score of 0.943. The obtained results are acceptable.

Furthermore, in Figure 13 we displayed words with least accuracy: Please, Angry, Friendly, Embarrassed and Soup. ASL word Angry was performed by clawing double hands and inserting fingertips against stomach. Then, hands were forcefully pulled up and outward. ASL word Please or Pleasure was performed by placing both hands on chest, with both palms facing outwards. Then, hands moved in circular motion. ASL word Friendly was performed by raising double hands a few inches in front of head. Then, fingers were wiggled using double hands backward movement. The low accuracy was due to word Please being misclassified as Angry and vice versa. Recognition of these words is thorny, because they share similar considerable parameters.

In addition, CMC curve is designed to illustrate recognition rate versus rank score. In this plot, each learning set exempted the knowledge of hand dynamics to measure the similarity contribution of each word combinations. Thus, best recognition was achieved at lower rank, whereas low recognition was achieved from the high rank, as shown in Figure 14. The double-hand ASL words with least ranks were Car (10th), Come (35th), Finish (14th), Go (29th) and Good (2nd). These gestures can achieve best recognition without knowledge of hand dynamics, whereas ranks 8th, 12th, 18th, 23rd and the remaining ranks are difficult to recognize without knowledge of hand dynamics. This demonstrates that not all gestures are unique; each gesture needs different number of discriminating features during recognition. It is worth noting that manual hand features are promising to address misclassification. However, it is difficult to design network suitable for all the dynamic hand gestures. To overcome this challenge, there is need to design network that has a series of concatenated classifiers, so that each group of gestures could have a suitable classifier, as well as features.



Recognition score: 0.95

Embarrassed

Recognition score: 0.92

Angry

Recognition score: 0.93

Soup

Recognition score: 0.91

Please

**Figure 13.** Misclassified words.

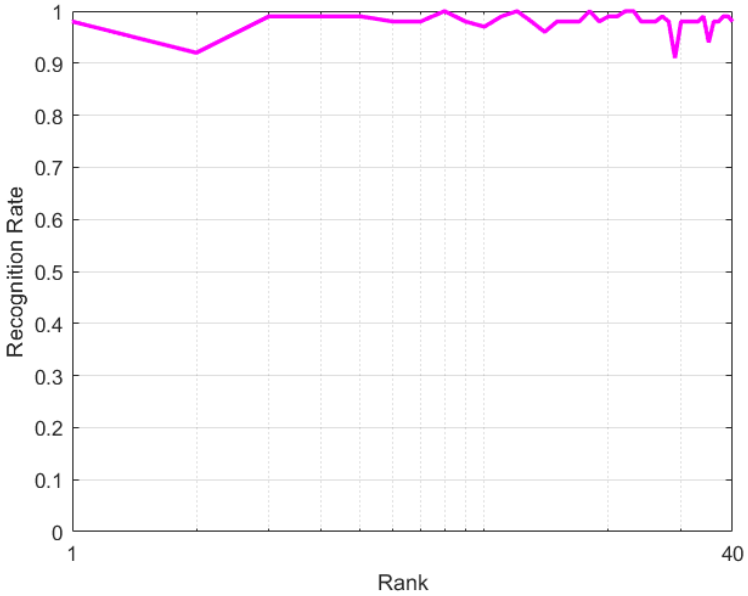


**Figure 14.** CMC evaluation.

## 5. Conclusions

In this work, we addressed the misclassification problem of double-hand dynamic similar and non-similar ASL words. The method achieved an average a recognition accuracy of 97.983% when aiding an effective and automatic recognition of complex double-hand dynamic ASL words from 3D skeletal hand-joint video features of hand motion trajectories and pause, which were developed inside a multi-stacked deep BiLSTM enhanced with machine learning tools. The proposed method designed a consolidated input feature vector. Our method outperformed the existing state-of-the-art methods. Although we experienced misclassification of a few words, it is worth emphasizing that multi-stacked deep BiLSTM initialized from transfer learning with multi-features is promising with regard to challenging, small and large vocabularies of static and dynamic sign words. In a nutshell, misclassification of double-hand dynamic gestures and general gestures could be addressed by extending the vocabulary to accommodate more gestures with various complexities. In addition, if we are to consider the real application of sign-language recognition, then the recognition network should be trained on a relatively small number of gestures, and recognition could be treated as a multi-feature problem. This work can be applied to ubiquitous SLR systems, mobile games, and robotics. Further research should investigate spatial information from skeletal hand-joint video frames to address the misclassification of dynamic sign words.

**Author Contributions:** Conceptualization, K.C. and S.B.A.; Formal analysis, S.B.A.; Funding acquisition, K.C.; Project administration, K.C.; Supervision, K.C.; Writing—original draft, S.B.A.; Writing—review and editing, S.B.A. and K.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research project is financially supported by Petchra Pra Jom Klao Ph.D. Research Scholarship (Grant No. 45/2562) from King Mongkut's University of Technology Thonburi Bangkok, Thailand.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in experiment.

**Data Availability Statement:** American sign language data set of dynamic words accessed (10 May 2021) at https://bitbucket.org/visionlab-sapienza/2018-jrl-ieee-tmm_-application_dataset/sr. LMDHG data set is accessed (13 May 2021) at https://www-intuidoc.irisa.fr/en/english-leap-motion-dynamic-hand-gesture-lmdhg-database/. SHREC'17 data set is accessed (13 May 2021) at http://www-rech.telecom-lille.fr/shrec2017-hand/.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Valli, C.; Lucas, C. *Linguistics of American Sign Language: An Introduction*; Gallaudet University Press: Washington, DC, USA, 2000.
2. Brentari, D. *Sign Languages*, 1st ed.; Cambridge University Press: Cambridge, UK, 2010; pp. 405–432.
3. Mitchell, R.E.; Young, T.A.; Bachelda, B.; Karchmer, M.A. How many people use ASL in the United States? Why estimates need updating. *Sign Lang. Stud.* **2021**, *6*, 306–335. [CrossRef]
4. Gokce, C.; Ozdemir, O.; Kindiro, A.A.; Akarun, L. Score-level Multi Cue Fusion for Sign Language Recognition. In Proceedings of the Lecture Notes in Computer Science, European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020.
5. Lee, C.K.; Ng, K.K.; Chen, C.H.; Lau, H.C.; Chung, S.; Tsoi, T. American sign language recognition and training method with recurrentneural network. *Expert Syst. Appl.* **2021**, *167*, 114403. [CrossRef]
6. Frishberg, N. Arbitrariness and iconicity: Historical change in American Sign Language. *Language* **1975**, *51*, 696–719. [CrossRef]
7. Liao, Y.; Xiong, P.; Min, W.; Min, W.; Lu, J. Dynamic sign language recognition based on video sequence with BLSTM-3D residual networks. *IEEE Access* **2019**, *7*, 38044–38054. [CrossRef]
8. Konstantinidis, D.; Dimitropoulos, K.; Daras, P. A Deep Learning Approach for Analyzing Video and Skeletal Features in Sign Language Recognition. In Proceedings of the 2018 IEEE International Conference on Imaging Systems and Techniques (IST), Krakow, Poland, 16–18 October 2018; pp. 1–6.
9. Rastgoo, R., Kiani, K.; Escalera, S. Hand sign language recognition using multi-view hand skeleton. *Expert Syst. Appl.* **2020**, *150*, 113336. [CrossRef]
10. Ye, Y.; Tian, Y.; Huenerfauth, M.; Liu, J. Recognizing american sign language gestures from within continuous videos. In Proceedings of the IEEE International Conference on Computer Vision Workshops, Venice, Italy, 22–29 October 2018.
11. Sharma, S.; Kumar, K. Asl-3dcnn: American sign language recognition technique using 3-d convolutional neural networks. *Multimed. Tools Appl.* **2021**, *2021*, 1–13. [CrossRef]
12. Mohandes, M.; Aliyu, S.; Deriche, M. Arabic sign language recognition using the leap motion controller. In Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey, 1–4 June 2014; pp. 960–965.
13. Nguyen, H.B.; Do, H.N. Deep learning for american sign language fingerspelling recognition system. In Proceedings of the 2019 26th International Conference on Telecommunications (ICT), Hanoi, Vietnam, 8–10 April 2019; pp. 314–318.
14. Naglot, D.; Kulkarni, M. Real time sign language recognition using the leap motion controller. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; pp. 314–318.
15. Chong, T.W.; Lee, B.G. American sign language recognition using leap motion controller with machine learning approach. *Sensors* **2018**, *18*, 3554. [CrossRef]
16. Chophuk, P.; Pattanaworapan, K.; Chamnongthai, K. Fist american sign language recognition using leap motion sensor. In Proceedings of the 2018 International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand, 7–9 January 2018; pp. 1–4.
17. Shin, J.; Matsuoka, A.; Hasan, M.; Mehedi, A.; Srizon, A.Y. American Sign Language Alphabet Recognition by Extracting Feature from Hand Pose Estimation. *Sensors* **2021**, *21*, 5856. [CrossRef]
18. Dutta, K.K.; Satheesh Kumar Raju, K.; Anil Kumar, G.S.; Sunny Arokia Swamy, B. Double handed Indian Sign Language to speech and text. In Proceedings of the 2015 Third International Conference on Image Information Processing (ICIIP), Waknaghat, India, 21–24 December 2015; pp. 589–592.
19. Demircioglu, B.; Bulbul, G.; Kose, H. Turkish sign language recognition with leap motion. In Proceedings of the 2016 24th Signal Processing and Communication Application Conference (SIU), Zonguldak, Turkey, 24–26 June 2020; pp. 589–592.
20. Mohandes, M.A. Recognition of two-handed arabic signs using the cyberglove. *Arab. J. Sci. Eng.* **2013**, *38*, 669–677. [CrossRef]
21. Haque, P.; Das, B.; Kaspy, N.N. Two-Handed Bangla Sign Language Recognition Using Principal Component Analysis (PCA) and KNN Algorithm. In Proceedings of the 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox's Bazar, Bangladesh, 7–9 February 2019; pp. 181–186.
22. Raghuveera, T.; Deepthi, R.; Mangalashri, R.; Akshaya, R. A depth-based Indian sign language recognition using microsoft kinect. *Sadhana* **2020**, *45*, 1–13. [CrossRef]
23. Karacı, A.; Akyol, K.; Turut, M.U. Real-Time Turkish Sign Language Recognition Using Cascade Voting Approach with Handcrafted Features. *Appl. Comput. Syst.* **2021**, *26*, 12–21. [CrossRef]
24. Katılmıs, Z.; Karakuzu, C. ELM Based Two-Handed Dynamic Turkish Sign Language (TSL) Word Recognition. *Expert Syst. Appl.* **2021**, *2021*, 115213. [CrossRef]
25. Kam, B.D.; Kose, H. A New Data Collection Interface for Dynamic Sign Language Recognition with Leap Motion Sensor. In Proceedings of the Game Design Education: Proceedings of PUDCAD 2020, Virtual Conference, 24–26 June 2020; Springer: Berlin/Heidelberg, Germany, 2021; pp. 353–361.
26. Hisham, B.; Hamouda, A. Arabic sign language recognition using Ada-Boosting based on a leap motion controller. *Int. J. Inf. Technol.* **2021**, *13*, 1221-1234. [CrossRef]

27. Fang, B.; Co, J.; Zhang, M. Deepasl: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, SenSys'17, Delft, The Netherlands, 5–8 November 2017; ACM: New York, NY, USA, 2017.
28. Masood, S.; Srivastava, A.; Thuwal, H.C.; Ahmad, M. Real-time sign language gesture (word) recognition from video sequences using cnn and rnn. *Intell. Eng. Inform.* **2018**, *2018*, 623–632.
29. Yang, L.; Chen, J.; Zhu, W. Dynamic hand gesture recognition based on a leap motion controller and two-layer bidirectional recurrent neural network. *Sensors* **2020**, *20*, 2106. [CrossRef]
30. Mittal, A.; Kumar, P.; Roy, P.P.; Balasubramanian, R.; Chaudhuri, B.B. A modified LSTM model for continuous sign language recognition using leap motion. *IEEE Sens. J.* **2019**, *19*, 7056–7063. [CrossRef]
31. Chophuk, P.; Chamnongthai, K. Backhand-view-based continuous-signed-letter recognition using a rewound video sequence and the previous signed-letter information. *IEEE Access* **2021**, *9*, 40187–40197. [CrossRef]
32. Avola, D.; Bernadi, L.; Cinque, L.; Foresti, G.L.; Massaroni, C. Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. *IEEE Trans. Multimed.* **2018**, *21*, 234–245. [CrossRef]
33. Itauma, I.I.; Kivrak, H.; Kose, H. Gesture imitation using machine learning techniques. In Proceedings of the 2012 20th Signal Processing and Communications Applications Conference (SIU), Mugla, Turkey, 30 May 2012.
34. Azar, G.; Wu, H.; Jiang, G.; Xu, S.; Liu, H. Dynamic gesture recognition in the internet of things. *IEEE Access* **2018**, *7*, 23713–23724.
35. Lupinetti, K.; Ranieri, A.; Giannini, F.; Monti, M. 3d dynamic hand gestures recognition using the leap motion sensor and convolutional neural networks. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Augmented Reality, Virtual Reality and Computer Graphics, Genova, Italy, 22–24 August 2020*; Springer: Berlin/Heidelberg, Germany, 2020.
36. Parelli, M.; Papadimitriou, K.; Potamianos, G.; Pavlakos, G.; Maragos, P. Exploiting 3d hand pose estimation in deep learning-based sign language recognition from rgb videos. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Cham, Switzerland, 2020; pp. 249–263.
37. Vaitkevicius, A.; Taroza, M.; Blazauskas, T.; Damasevicius, R.; Maskeliūnas, R.; Wozniak, M. Recognition of American sign language gestures in a virtual reality using leap motion. *Appl. Sci.* **2019**, *9*, 445. [CrossRef]
38. Igari, S.; Fukumura, N. Sign language word recognition using via-point information and correlation of they bimanual movements. In Proceedings of the 2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA), Bandung, Indonesia, 20–21 August 2014; pp. 75–80.
39. Aliyu, S.; Mohandes, M.; Deriche, M. Dual LMCs fusion for recognition of isolated Arabic sign language words. In Proceedings of the 2017 14th International Multi-Conference on Systems, Signals and Devices (SSD), Marrakech, Morocco, 28–31 March 2017; pp. 310–321.
40. Deriche, M.; Aliyu, S.O.; Mohandes, M. An intelligent arabic sign language recognition system using a pair of LMCs with GMM based classification. *IEEE Sens. J.* **2019**, *19*, 8067–8078. [CrossRef]
41. Katılmıs, Z.; Karakuzu, C. Recognition of Two-Handed Posture Finger Turkish Sign Language Alphabet. In Proceedings of the 2020 5th International Conference on Computer Science and Engineering (UBMK), Diyarbakir, Turkey, 9–11 September 2020; pp. 181–186.
42. Goldin-Meadow, S.; Brentari, D. Gesture, sign, and language: The coming of age of sign language and gesture studies. *Behav. Brain Sci.* **2017**, *2017*, 1–60. [CrossRef]
43. Abdullahi, S.B.; Chamnongthai, K. American Sign Language Words Recognition using Spatio-Temporal Prosodic and Angle Features: A sequential learning approach. *IEEE Access* 2022, *in press*. [CrossRef]
44. Huber, P.J. The Basic Types of Estimates. In *Robust Statistics*; A John Wiley and Sons, Inc.: Hoboken, NJ, USA, 2004; pp. 43–68.
45. Song, C.; Zhao, H.; Jing, W.; Zhu, H. Robust video stabilization based on particle filtering with weighted feature points. *IEEE Trans. Consum. Electron.* **2012**, *58*, 570–577.
46. Kiani, M.; Sim, K.S.; Nia, M.E.; Tso, C.P. Signal-to-noise ratio enhancement on sem images using a cubic spline interpolation with savitzky–golay filters and weighted least squares error. *J. Microsc.* **2015**, *258*, 140–150. [CrossRef]
47. Balcilar, M.; Sonmez, A.C. Background estimation method with incremental iterative re-weighted least squares. *Signal Image Video Process.* **2016**, *10*, 85–92. [CrossRef]
48. Ma, J.; Zhou, Z.; Wang, B.; Zong, A.C. Infrared and visible image fusion based on visual saliency map and weighted least square optimization. *Infrared Phys. Technol.* **2017**, *82*, 8–17.
49. Xia, K.; Gao, H.; Ding, L.; Liu, G.; Deng, Z.; Liu, Z.; Ma, C. Trajectory tracking control of wheeled mobile manipulator based on fuzzy neural network and extended Kalman filtering. *Neural Comput. Appl.* **2018**, *30*, 447–462. [CrossRef]
50. Reshef, D.N.; Reshef, Y.A.; Finucane, H.K.; Grossman, S.R.; McVean, G.; Turnbaugh, P.J.; Lander, E.S.; Mitzenmacher, M.; Sabeti, P.C. Detecting novel associations in large data sets. *Science* **2011**, *6062*, 1518–1524. [CrossRef]
51. Zhang, Y.; Jia, S.; Huang, H.; Qiu, J.; Zhou, C. A novel algorithm for the precise calculation of the maximal information coefficient. *Sci. Rep.* **2014**, *1*, 6662. [CrossRef]
52. Li, H.; Wu, L.; Wang, H.; Han, C.; Quan, W.; Zhao, J. Hand gesture recognition enhancement based on spatial fuzzy matching in leap motion. *IEEE Trans. Ind. Informatics* **2019**, *16*, 1885–1894. [CrossRef]
53. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1999**, *45*, 2673–2681. [CrossRef]
54. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]

55. Keren, G.; Schuller, B. Convolutional rnn: an enhanced model for extracting features from sequential data. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, Canada, 24–29 July 2016.
56. Rastgoo, K.; Kiani, K.; Escalera, S. Real-time isolated hand sign language recognition using deep networks and svd. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *16*, 1–21. [CrossRef]
57. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
58. Bishop, C.M. Pattern recognition. *Mach. Learn.* **2006**, *4*, 738.
59. Mocialov, B.; Turner, G.; Hastie, H. Transfer learning for british sign language modelling. *arXiv* **2020**, arXiv:2006.02144.
60. Bird, J.J.; Ekrt, A.; Faria, D.R. British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language. *Sensors* **2020**, *20*, 5151. [CrossRef]
61. Chicco, D.; Jurman, G. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 6. [CrossRef]
62. Chicco, D.; Tosch, N.; Jurman, G. The matthews correlation coefficient (mcc) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Min.* **2021**, *14*, 1–22. [CrossRef]
63. Depaolini, M.R.; Ciucci, D.; Calegari, S.; Dominoni, M. External Indices for Rough Clustering. In *Rough Sets, Lecture Notes in Computer Science, Proceedings of the International Joint Conference on Rough Sets (IJCRS) 2018, Quy Nhon, Vietnam, 20–24 August 2018*; Nguyen, H., Ha, Q.T., Li, T., Przybyła-Kasperek M., Eds.; Springer: Berlin/Heidelberg, Germany, 2018.
64. Boulahia, S.Y.; Anquetil, E.; Multon, F.; Kulpa, R. Dynamic hand gesture recognition based on 3D pattern assembled trajectories. In Proceedings of the 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, QC, Canada, 28 November–1 December 2017.
65. De Smedt, Q.; Wannous, H.; Vandeborre, J.P. 3d hand gesture recognition by analysing set-of-joints trajectories. In Proceedings of the International Workshop on Understanding Human Activities through 3D Sensors, Cancun, Mexico, 4 December 2018; Springer: Cham, Switzerland, 2018.
66. Chui, K.T.; Fung, D.C.L.; Lytras, M.D.; Lam, T.M. Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Comput. Hum. Behav.* **2020**, *107*, 105584. [CrossRef]
67. Li, Y.; Ma, D.; Yu, Y.; Wei, G.; Zhou, Y. Compact joints encoding for skeleton-based dynamic hand gesture recognition. *Comput. Graph.* **2021**, *97*, 191–199. [CrossRef]
68. Liu, J.; Liu, Y.; Wang, Y.; Prinet, V.; Xiang, S.; Pan, C. Decoupled representation learning for skeleton-based gesture recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5751–5760.
69. Boulahia, S.Y.; Anquetil, E.; Kulpa, R.; Multon, F. HIF3D: Handwriting-Inspired Features for 3D skeleton-based action recognition. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016.
70. Ohn-Bar, E.; Trivedi, M. Joint angles similarities and HOG2 for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, OR, USA, 23–28 June 2013.
71. Oreifej, O.; Liu, Z. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Portland, OR, USA, 23–28 June 2013.
72. Devanne, M.; Wannous, H.; Berretti, S.; Pala, P.; Daoudi, M.; Del Bimbo, A. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE Trans. Cybern.* **2014**, *45*, 1340–1352. [CrossRef]
73. Hou, J.; Wang, G.; Chen, X.; Xue, J.H.; Zhu, R.; Yang, H. Spatial-temporal attention res-TCN for skeleton-based dynamic hand gesture recognition. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018.
74. Chen, X.; Wang, G.; Guo, H.; Zhang, C.; Wang, H.; Zhang, L. Mfa-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data. *Sensors* **2019**, *19*, 239. [CrossRef]
75. Caputo, F.M.; Prebianca, P.; Carcangiu, A.; Spano, L.D.; Giachetti, A. Comparing 3D trajectories for simple mid-air gesture recognition. *Comput. Graph.* **2018**, *73*, 17–25. [CrossRef]
76. Maghoumi, M.; LaViola, J.J. DeepGRU: Deep gesture recognition utility. In Proceedings of the International Symposium on Visual Computing, Nevada, CA, USA, 7–9 October 2019; Springer: Cham, Switzerland, 2019.