International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 6
ISSN: 2088-5334

Dynamic Sign Language Recognition Using Mediapipe Library and Modified LSTM Method

Ridwang [a,*], Amil Ahmad Ilham [b], Ingrid Nurtanio [b], Syafaruddin [c]

[a] Department of Electrical Engineering, Universitas Muhammadiyah Makassar, Makassar 90221, Indonesia
[b] Department of Informatics, Universitas Hasanuddin, Gowa, 92171, South Sulawesi, Indonesia
[c] Department of Electrical Engineering, Universitas Hasanuddin, Gowa,, 92171, South Sulawesi, Indonesia
Corresponding author: *ridwang@unismuh.ac.id

Abstract— **Hand gesture recognition (HGR) is a primary mode of communication and human involvement. While HGR can be used to enhance user interaction in human-computer interaction (HCI), it can also be used to overcome language barriers. For example, HGR could be used to recognize sign language, which is a visual language expressed by hand movements, poses, and faces, and used as a basic communication mode by deaf people around the world. This research aims to create a new method to detect dynamic hand movements, poses, and faces in sign language translation systems. The Long Short-Term Memory Modification (LSTM) approach and the Mediapipe library are used to recognize dynamic hand movements. In this study, twenty dynamic movements that match the context were designed to solve the challenge of identifying dynamic signal movements. Sequences and image processing data are collected using MediaPipe Holistic, processed, and trained using the LSTM Modification method. This model is practiced using training and validation data and a test set to evaluate it. The training evaluation results using the confusion matrix achieved an average accuracy of twenty words trained, which was 99.4% with epoch 150. The results of experiments per word showed detection accurateness of 85%, while experiments using sentences only reached 80%. The research carried out is a significant step forward in advancing the accuracy and practice of the dynamic sign language recognition system, promising better communication and accessibility for deaf people.**

Keywords— **Deaf people; modification LSTM; static sign; words sign; sentence sign.**

Manuscript received 15 Aug. 2023; revised 31 Oct. 2023; accepted 17 Nov. 2023. Date of publication 31 Dec. 2023.

I. INTRODUCTION

Humans use sign language as a means of communicating with normal people. Ordinary sign language combines face and hand movements [1]. Face expressions, motions of the lips, and head movements are examples of non-manual signals, while hand and finger movements, orientation of the hands, and movement of the body are examples of manual signals [2]. The use of sign language for communication varies from country to country and has no standardization. Unlike the oral language, where one word appears after another, the structure of sign language changes regarding geographical information [3]. However, as stated, a sign language phrase often comprises time, place, individual, and predicate.

According to Tan et al. [4], hand gestures are considered a natural and fundamental human interaction and communication tool, as they have been used to send information even before the advent of language. With a series of hand gestures and finger placements, complicated tasks can be easily done, and information conveyed. As a result, hand gestures can be employed as a highly flexible interface for human-computer interaction (HCI), facilitating faster interaction by removing the need for users to contact the mechanical device physically.

Furthermore, hand gestures are the primary way of communication for people who are deaf or have hearing loss [4], [5]. They were used as an implicit embodiment metric with communicative gestures [6]. In essence, hand movements serve as sign language [7]. Communication with the general populace can be difficult for those who are deaf or have hearing loss. One must learn sign language to comprehend the meaning of hand gestures used in both professional and informal communication. Because it may be utilized to overcome communication hurdles, developing a hand gesture detection system (SLR) is essential.

Sign language prediction aims to provide an auxiliary system that automatically translates the input signal into the

correct text or pronunciation. The SLR system is highly helpful in bridging the communication gap between society and fools. As a result, the technology opens new possibilities for applications that rely on human-computer interaction (HCI). Several effective SLR techniques for words have been developed by researchers [5]. Although isolated, words cannot understand and convey the sequence of ongoing movements. A major challenge in developing an SLR system that can be sustained is finding a model paradigm that can capture the appropriate signals and language. This problem is solved with voice recognition [8].

Think about language modeling using phonetic units that appear in sequence. However, the same idea can be used in a continuous SLR approach, where the sequence of signals, not words or phonemes, is available. A continuous SLR system for American Sign Language (ASL) sentences was created, employing three orthogonally cameras to address the issues brought on by occlusion and unrestricted movement. The Hidden Markov Model, which has a treasury of 53 signals, is used in identification procedures. The system recognition rate on 97 sentence signals with and without bigram modeling was 92.11%, respectively. A similar approach that uses a single camera and a 40-signal vocabulary was developed by Hinchcliffe et al. [9] for the word of the word. HMM classification is used in this situation to perform the identification process. However, SLR systems created using a single video camera have issues since (a) signature fingers and hand movements are obscured, (b) the signature is not always in front of the camera, and (c) there is a loss of depth due to the nature of a single 2D camera [10].

The employment of depth-supporting sensors, such as Leap Motion, improves the comprehension of input data [11] and Microsoft Kinect [12], which offers a 3D point cloud of the field seen. While Leap Motion tracks finger and hand movements in real-time, Kinect interprets body movements. This gadget helps solve problems with the SLR systems recorded by the previously mentioned 2D video cameras. In this work, we have developed continuous SLRs with independent modeling of each signal movement using 3D hand and finger movement data collected from the Leap motion sensor. In most continuous SLR systems, HMM implicitly segmented the signal sequence into the composer signal[13]. The Markov chain relies on a fixed window. This chain develops and recognizes characters from character history using cutting-edge machine learning techniques such as simulated neural networks (RNN) [14].

However, no studies above show a method for detecting and deleting transition signals in dynamic signal languages. It can improve the accuracy of dynamic sign language detection. As a result, the main contribution of this research is as follows.

- The Proposed method can detect and remove transition signals in dynamic sign language.
- The proposed method can be applied to detect signals from hands, poses, and faces in dynamic signal language with high accuracy and a rapid data extraction process.

In this study, MediaPipe Holistic uses hand, position, and face expression landmark models to generate 543 landmarks, including 33 pose landmarks, 42 hand landmarks, and 468 face landmarks. The following section of this essay is organized as follows. Part I provides a broad overview of the most recent research on continuous SLR. Part II provides a detailed description of the pre-processing, feature extraction, training, and testing processes involved in the implementation method we provide for continuous SLR systems. In Part III, we discuss datasets and compile information from numerous studies. The part offers suggestions, findings, and possible areas for more study.

II. Material and Method

The foundation of the current SLR approach includes 2D video cameras, color gloves, sensor gloves etc. [15]. SLR research has recently focused on new 3D environments that use cameras and deep sensors [12]. HMM, Hypothesis Neural Network (JST), RNN, and rule-based modeling techniques are the foundation of most signal recognition work. Anand et al. [16] developed a system that continuously used a single video camera for view-based methodology. In the first stage, a camera mounted on a disk views the signature, and in the second stage, a camera attached to the hat views the signature [17]. Hand segmentation and hand clamp extraction are performed using a vision-based skin color modeling technique. The authors utilized hand clump analysis to consolidate the position, angle, breadth, and length features into a 16-dimensional feature vector.

For devices mounted on tables and systems attached to headgear, training is conducted on 384 and 400 ASL sentences, respectively. Using HMM and 40 signal words, the system was evaluated on 94 and 100 ASL phrases, with accuracy measured at 74.5% and 97.8% for table-based and heat-based systems, respectively. The authors of [18] have suggested a video camera-based continuous SLR For a signer-independent approach for identifying single-handed static and dynamic gestures, double-handed static gestures, and finger-spelled Indian Sign Language (ISL) phrases from live Video. Pre-processing, Feature Extraction, and Classification are the three critical phases of the gesture recognition module.

Using skin color segmentation, the indicators are taken from a real-time video during the pre-processing stage. After the co-articulation removal phase, a suitable feature vector is extracted from the gesture sequence. Support Vector Machines (SVM) are then utilized to classify the retrieved features. The algorithm correctly identified single-handed dynamic words with 89% accuracy and finger spelling alphabets with 91% accuracy. Selfie-captured sign language video [19] is processed using only a smartphone's computer capability. A sign language feature space is created using video frames' pre-filtering, segmentation, and feature extraction. Repeatedly trained and tested classifiers on the sign feature spacing Minimum Distance and Artificial Neural Networks [20]. The power of the Sobel Edge Operator is increased by morphology and adaptive thresholding, resulting in nearly flawless segmentation of the hand and head sections that account for the minute vibrations of the selfie stick. The proposed technique performs well with an average Word Matching Score (WMS) of about 85.58% for MDC and 90% for ANN and a modest variance of 0.3 s in classification times.

For many years, Sreemathy et al. [21] have investigated the study of sign language recognition systems utilizing a variety of image processing and artificial intelligence approaches. However, the critical problem is to close the communication

gap between persons with disabilities and the general population. This study suggests a Python-based framework for classifying 80 sign language words. The models You Only Look Once version 4 (YOLOv4) and Support Vector Machine (SVM) with media-pipe have been proposed in this work. The linear, polynomial, and Radial Basis Function (RBF) kernels are all used in SVM. The system does not require additional pre-processing and image enhancement activities. The self-created picture collection used in this study has a total of 676 photos of 80 static signs. SVM with media-pipe has an accuracy of 98.62%, while YOLOv4 has an accuracy of 98.8%, both higher than the current state-of-the-art techniques.

A SLR system for Arabic Sign Language has been proposed by Almasre et al. [22]. It is not yet being done to examine how sensors can be used and how natural user interfaces can help with ArSL interpretation. Previous studies have demonstrated that a one-size-fits-all classifier modeling approach is unsuitable for all hand gesture recognition tasks. Therefore, this study investigated the optimal combination of algorithms with varying parameters and applied them alongside a sensor device to achieve the highest accuracy in recognizing American Sign Language (ArSL) gestures within a recognition system. The study proposed utilizing a dynamic prototype model (DPM) employing the Kinect sensor to identify specific dynamic ArSL gestures. The DPM applied eleven SVM, RF, and KNN predictive models with various parameter values. The study's results showed that the SVM models with a linear kernel and a cost parameter of 0.035 had the highest recognition accuracy rates for dynamically gestured words.

The authors developed the Arabic Sign Language recognition system using a similar data-based, handle-based methodology for 40 signed sentences that were classified using a modified version of the k-NN algorithm [22]. However, the method of conducting the evaluation varies depending on the user. The authors suggest a scalable Hidden Markov Model (HMM) approach for continuous Sign Language Recognition (SLR), which involves training a single universal transition model and includes further investigations based on digital gloves [11]. Cerna et al. [12] suggested a Kinect-based continuous SLR framework. To calculate the likelihood of HMM, the authors have also suggested low-complexity development procedures.

The Kinect SDK is used to extract six 3D frame features. When tested with HMM-based classification, the system had an error rate of 12.20% on 100 CSL sentences with 21 signal words. Recently, the authors of [10] suggested calibrating Kinect and Leap Motion sensors for ISL signal detection using CNN [23]. While capturing 50 separate ISL signals, they tracked the 3D positions of the fingers and hand movements. HMM classification is used to extract angle features for identification. Similar 3D text segmentation and recognition techniques were employed by [24] and [, who employed a Leap motion sensor. The authors captured 3D sentences above the Leap Motion Display Field in the air.

This section outlines our LSTM-based neural network architecture for using a camera for continuous Sign Language Recognition (SLR). Figure 1 shows the flow diagram system SLR, where the camera is utilized to gather with Mediapipe for sign inputs. Then, the training process uses the Modified LSTM Method from vector data. This system's output is the word of the classification that will be combined into a sentence.

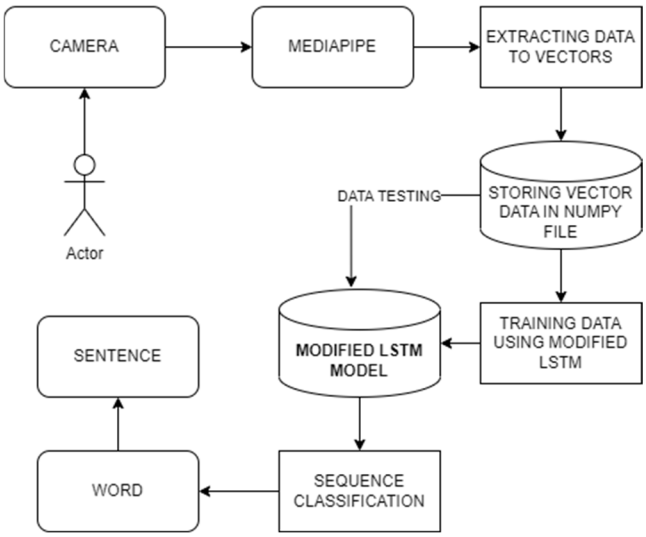


Fig. 1 Proposed diagram system for continuous SLR using Mediapipe and Modified LSTM

A. *Pre-processing*

Pre-processing is correcting images from camera detection results and performing feature extraction. All pre-processing processes were carried out in this study using the Mediapipe library. In the process, five stages must be passed: video conversion to frame, changing color transformation from BGR to RGB, Pose Detection, C cropping, and feature extraction[25].

1) Converting Video to frames: The conversion of Video to frames is a stage for creating a new image from the video data already taken. Each Video produces 30 frames according to the number of loops that have been determined.

2) Color Transfer from BGR to RGB: OpenCV detects images in the BGR format, so before pre-processing images, it is necessary to convert from the type of BGR to RGB using the functions of the OpenCV library.

3) Detection of poses: Figure 2 shows the Mediapipe model performing the first position prediction with the BlazePose detector and the following Landmark Models. After pose detection, three areas of Rest of Interest (ROI) plants were obtained - for two hands and faces, respectively.

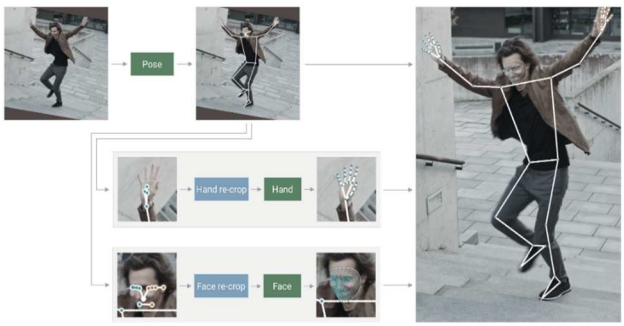


Fig. 2 Position hands, Pose, and face detection display.

For cases where the accuracy of the pose model is sufficiently low that the ROI generated for the hand is inaccurate, then the cropping model is run for the hands as a spatial transformation and only spends 10% of the hand model inference time.

4) Cropping: Figure 3 shows the cropping process carried out by the Mediapipe model by cropping against the ROI of the face and hand so that the model is more focused on the object of his face and hands.

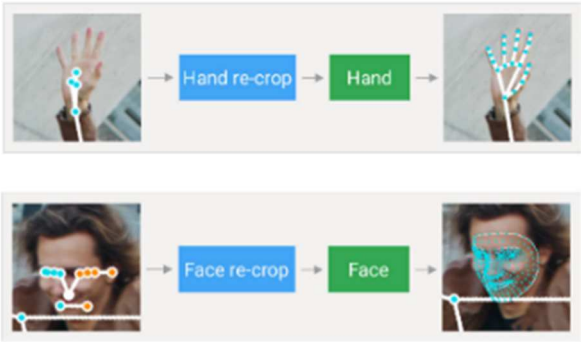


Fig. 3 Re-crop hands and face

B. Feature Extraction

The final step is to extract features from poses with 33 landmarks or keypoints and hands with 21 keypoints per hand. The result of the feature extraction from the pose is the amount of data obtained, which is as many as 33 keypoints multiplied by 4, which is as much as 132 data points [26].

- x and y express width and height
- z represents depth. (deep)
- Visibility has values [0,0 and 0,1] used to indicate whether the image is visible or not.

To extract the hand features, 21 key points were multiplied by 3 variables to produce 63 data points per hand. So, for two hands, generate as much as 126 data output extractions. While extracting features from the face produces keypoints of 468 with 3 dimensions: x, y, and z, the total data obtained is 1404 [27].

For a system design that uses input data poses and hand data extraction results, these are made into one dimension using the flat function on Tensorflow so that the extraction output is one-dimensional data. After extracting the feature data, the keypoints pose and hand are merged using the String Concatenate function in the Python library. The combination of poses and hands has 258 key points.

For a system design that uses input data such as poses, hands, and faces, 1662 keypoints were obtained from 132 poses, 126 hands, and 1404 faces. This number of keypoints will later be used as input into the LSTM network for the training process [28].

C. Modified LSTM

A modified LSTM is shown below. That is expected to reset the memory in the memory cell. The idea comes from the method, variant of the LSTM [28], [29], the GRU. Gate Recurrent Unit [30] has two gates: a reset gate and an update gate. This reset gate removes the influence of the previous time so that it will not affect the output in the future [31].

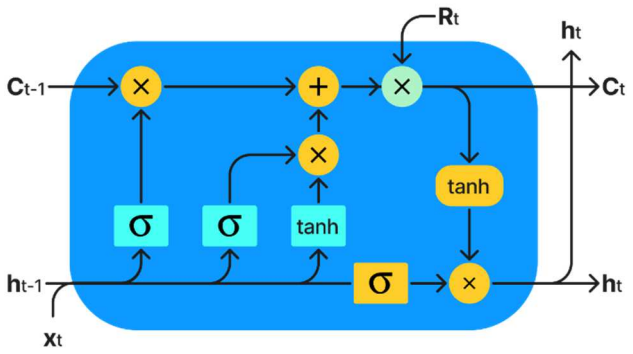


Fig. 4 Architecture Modified LSTM

Figure 4 shows the addition of the variable R_t, which is the reset variable at a certain time interval; the value of this variable is only two, 0 and 1. If the value is sent as a number zero, then the Cell state (C_t) and Hidden State (h_t) values become zero with the addition of a multiplication function like in the formula $C_{t\_new}=C_{\_t}*R_{\_t}$. If the value $R_{\_t} = 0$, then the value C_{t_new} becomes zero. Thus, C_t value becomes zero so that C_t memory value does not influence the next LSTM unit. The process of making zero hidden state values can be seen in the complete formula below.

$$f_t = \sigma\left(W_f.\left[h_{t-1,}x_t\right] + b_i\right) \quad (1)$$

$$i_t = \sigma\left(W_i.\left[h_{t-1,}x_t\right] + b_i\right) \quad (2)$$

$$\hat{C}_t = tanh\left(W_c.\left[h_{t-1,}x_t\right] + b_c\right) \quad (3)$$

$$C_t = \left(f_t * C_{t-1} + i_t * \hat{C}_t\right) \quad (4)$$

$$C_{t_{new}} = C_t * R_t \quad (5)$$

$$O_t = \sigma\left(W_o.\left[h_{t-1,}x_t\right] + b_o\right) \quad (6)$$

$$h_t = O_t * tanh\left(C_{t_{new}}\right) \quad (7)$$

The addition of the multiplication function above is intended to make changes in the value of the cell state at the time (t), so that the value from ht becomes zero with the formula ht, which is to multiply the output gate O_t value by tanh $C_{t\,new}$. The value of C_t is equal to zero, so tanh (0) is zero, and the value of ht is also zero because the combination of the output gate value with the number zero will produce a zero. By making the values of C_t and ht zero so that the influence of the value does not exist on the next LSTM unit and by giving good accuracy to the output of the nerve network.

For the entire system process, parameters are added using the LSTM method, i.e., the parameters r and t. The parameter r is used to send the value of the output of the SoftMax layer, which is zero or one. The t value is the time during which the process occurs in the nerve network. The t value plays a very important role because of the cell state and hidden state values in time-based reset (t). Then add the formula to find the new C_t value in the LSTM method that uses the function of the TensorFlow Hard Framework.

By modifying the LSTM method and the output value of the SoftMax activation function, the influence of the transition movement, which previously became noise, can be eliminated and can increase the accuracy of the signal language translation. In addition to the above modification, the next challenge is detecting the precise transition movement so that it can remove the cell state value at the correct time (t).

D. Training and Testing

Our model is composed of three layers of modified LSTM units with Rectified Linear Unit (ReLU) activation, followed by a dense layer, and culminating in a SoftMax layer for classifying into multiple categories. We employed the Adadelta optimizer, known for its resilience against noisy gradients and its avoidance of the need for manual learning rate adjustments. The model is initially trained using discrete sign motions for effective framework modeling and then polished with continuous gestures. Continuous signs are produced from discrete motions by adding a variable length transition between states or gestures.

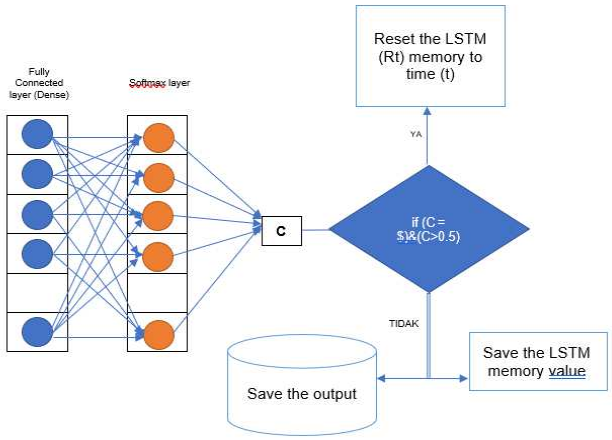


Fig. 5 Modification diagram of output Softmax

Figure 5 describes the output of the SoftMax activation function where this activation functions all the values to the existing target class with the sum of the probability values for all classes up to one. Each output of a vector number is translated into a target class in the form of a string. The output layer SoftMax as the activation function of the output Layer issues or distributes probability values to all existing target classes. For example, if the target class has 20 words/label, then will be 20 values removed from this activation function with the total of the total values being one. The process often done on the previous research is that each output of the SoftMax is directly translated to the respective target class to be stored in the form of a model.

In this study, modifications or additions of algorithms were made to the output of SoftMax or output from a nerve network containing several layers of LSTM and Dense. This modification is used to detect features that do not influence the output or other words that do not affect the output value. Each feature that passes through the network and the C value that comes out of the SoftMax layer and is not part of the transition label will then be forwarded to the word storage in the form of a string that will subsequently become a model. If the C value that comes out of the SoftMax activation function is a part or association of the transition label with a probability value C > 0.5 then it will call the reset function to perform the removal of the value on the LSTM memory so as not to affect the next frame that will enter the network. For more clarity you can see the algorithm of the modification of the output layer SoftMax below:

- The target class has 20 words/label.
- On the output layer SoftMax distributes values into 20 target classes.
- Value C is the value of the distribution result for the entire target class.
- If the value C = tr and C > 0,5, then it is categorized as a transition label; otherwise, its value will be forwarded to be stored as a target label.

The above stages describe the process of training using six layers with the activation function of SoftMax on the output layer. The variable value C is the probability value of SoftMax against the existing target class. If the value C is equal to or associated with the transition label (tr) and the probability value C > 0.5, then the system categorizes it as a transition feature that will subsequently call the reset function on the LSTM. A value of 0.5 is the minimum value used as a threshold to determine whether a feature is categorized as a transition or not. Previous experiments have also been conducted at the thresholds of 0.2, 0.3, and 0.4, showing results that are too sensitive to transition feature readings. This is influenced by the presence of several word signal frames that are like transition signal frames. The same experiment was once conducted experiments on the probability values of 0.6 and 0.7. The result obtained was less sensitive reading features of the signal, even more so with the number of target classes, so the probabilities to each target class were small.

III. Results and Discussion

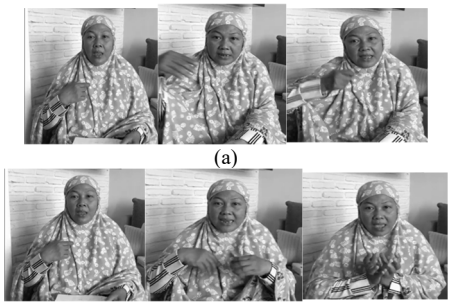
Here, we give specifics on the datasets that were captured using the suggested SLR methodology. The continual identification of sign language has then been given. The outcomes of the word recognition of solitary signs have finally been presented. Four people have signed up to participate in the collection of sign language data. Two of the participants were young people who attended a hearing-impaired school in Gowa, Indonesia. They are from the Indonesian city of Makassar's Community Deaf. 20 solitary signs in words make up the dataset. Each signatory has said each sign word at least 30 times. As a result, 2400 sign words (20*30*4) in all are noted. In Table I, you'll find a comprehensive description of all the sign words, with the '$' sign indicating transitional movements, representing the switch between two continuous letters within a sign sentence. Twenty examples of dynamic gesture visualization are presented to justify this model. They were selected based on the words most frequently used by deaf people, namely: "Love", "Everything", "Ball", "Rice", "There", "Remember", "They"," Market"," You"," Hear"," Play"," Go"," See"," Laugh"," We"," Honestly"," Cry"," Sleep"," Cook" and "$". Table I shows visualizations of the selected words.

TABLE I
WORDS SIGN LANGUAGE IN THE DATASET

Love	Everything	Ball	Rice
There	Remember	They	Market
You	Hear	Play	Go
See	Laugh	We	Honestly
Cry	Sleep	Cook	$

In Table 1, twenty words in the dataset represent the categories of nouns, verbs, adjectives, and adverbs. In addition, the words selected in the data set are the most frequently used words by deaf people in their daily activities. The purpose of the word selection is to be able to make a

sentence that is intact for the deaf at the time of the test. In Figure 6, an illustration of a dynamic signal on a sentence sample is provided.



(a)

(b)

Fig. 6 Example signals in the form of a sentence (a) we go to the market (b) we play the ball.

Figure (a) shows the word "Go", depicted using one hand as if throwing something away, then shows the word "Market", where the movement of the right hand gives a movement as if holding money and then depicts the word "We", this is shown by the right hand pointing at the person. Picture (b) depicts the word "We". This is shown by the right hand pointing at the person and then depicting the "Play" word, where the left and right hands movements are rotated together, and then the word "Ball". The word ball is described using two hands as if holding a round object.

The results of the analysis of the 20 classes in the dataset will be displayed in Figure 7, which displays the confusion matrix. The analysis of 20 classes based on the TP, TN, FP, and FN confusion matrix data will produce each class's accuracy, sensitivity, and specificity using the LSTM model.

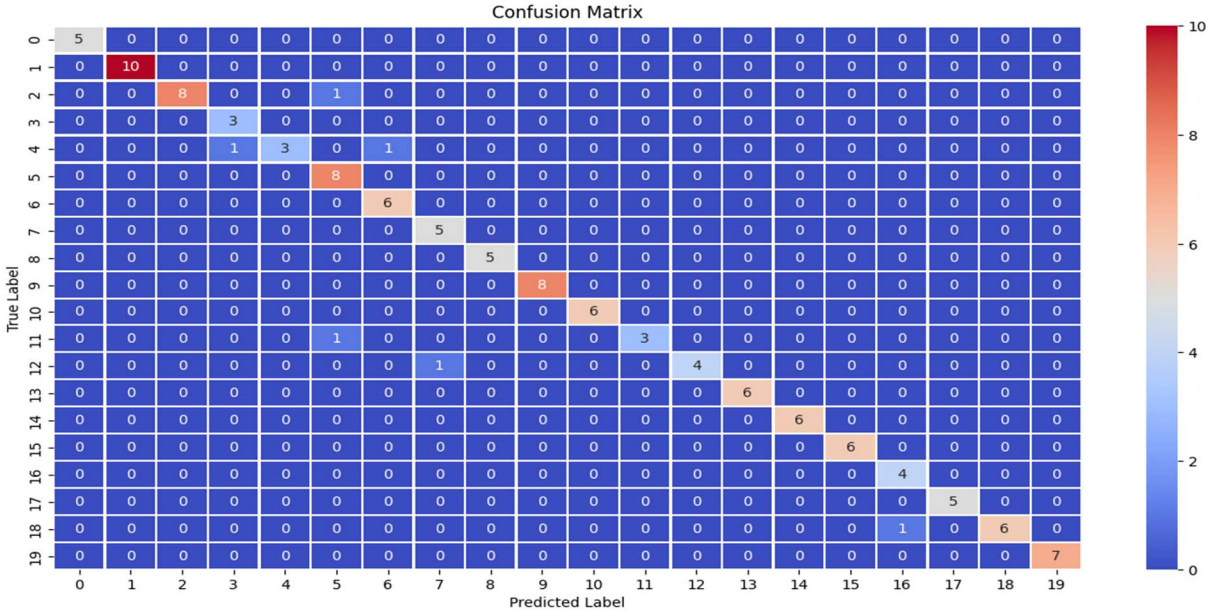


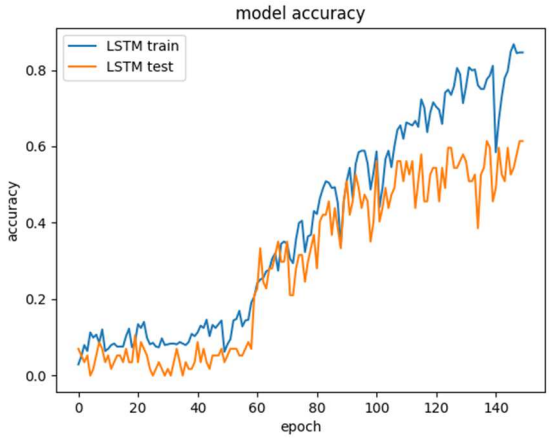
Fig. 7 Confusion Matrix Modified LSTM

TABLE II
AVERAGE ACC, SE AND SP FOR 20-CLASES

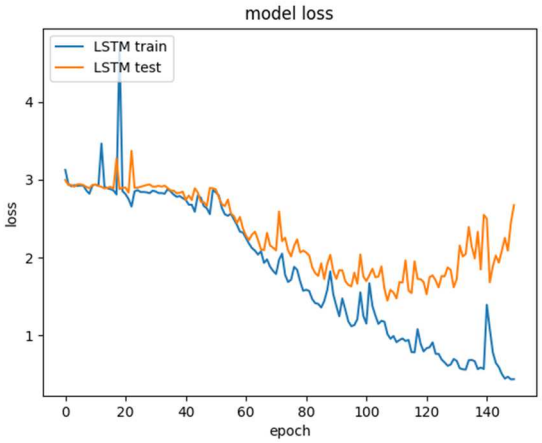
Words	Accuracy (%)	Sensitivity (%)	Specificity (%)
Love	100	100	100
There	100	100	100
You	99	89	100
See	99	100	99
Cry	98	60	100
Everything	98	100	98
Remember	99	100	99
Hear	99	100	99
Laugh	100	100	100
Sleep	100	100	100
Ball	100	100	100
They	99	75	100
Play	99	80	100
We	100	100	100
Cook	100	100	100
Rice	100	100	100
Market	99	100	99
Go	100	100	100
Honestly	99	86	100
$	100	100	100

To display the analysis results of twenty classes in the dataset, calculate each class's accuracy, sensitivity, and specificity values. Table II shows the calculations for twenty different classes' accuracy, sensitivity, and specificity value calculations. Based on the confusion matrix analysis, the accuracy results are quite impressive. Out of a total of 20 words, ten words achieved a perfect 100 percent accuracy rate. These words are Love, there, Laugh, Sleep, Ball, We, Cook, Rise, Go, and the special character "$" (transition sign). In addition, eight words reached accuracy 98: You, See, Remember, Hear, They, Play, Market, and Honestly. Among the words analyzed, Cry, and Everything had the lowest accuracy values, both performing at a commendable 98 percent accuracy rate. The challenges these words face in achieving perfection may be attributed to the quality of the training data, as suboptimal data often impact lower-performing words during the training process.

The graphical representation of accuracy and loss for a dataset comprising 20 different classes is visually depicted in Figure 8. This training process, conducted over 150 epochs, demonstrates impressive efficiency, with completion times as short as 10 minutes when employing the LSTM and Mediapipe methods.



(a)



(b)

Fig. 8 Accuracy (a) and Loss (b) Model on Epoch 150

A. *Evaluations using Signed Sentences*

To demonstrate its ability to recognize signed sentences, the enhanced LSTM model was trained on sign language words. The network was trained using a categorical cross-entropy loss function, incorporating 256 hidden states within each LSTM layer, a batch size of 256, an adaptive learning rate initialized at 0.001, and these specified parameters. Training took place over 150 epochs, and the entire process was completed within 10 minutes using an NVIDIA Intel GPU workstation, with a RESET LSTM state condition applied for transition gestures ($).

Based on the results of testing 320 sentences, which consist of 6 categories of sentences containing two to six phrases each, drawn from 8 sources, the highest average accuracy achieved was 84%, while the lowest was 77%. The highest accuracy was obtained in the category of 2 phrases, and the lowest accuracy was observed in the category of 6 phrases within each sentence. In Figure 9, a graph displaying the accuracy of sentences in each category is presented. It is evident that sentences with fewer phrases have higher accuracy compared to sentences with a greater number of phrases.

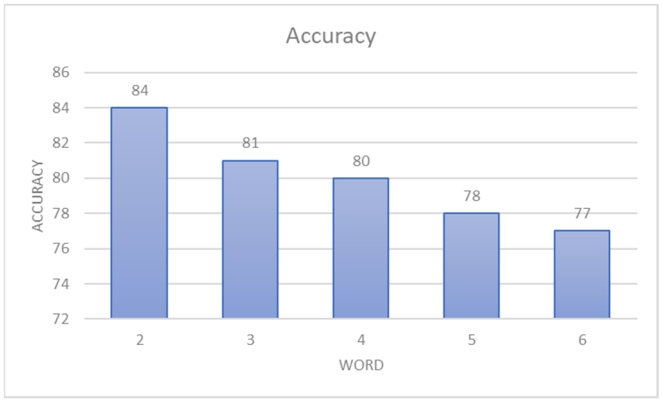


Fig. 9 Recognition accuracies of signed sentences

The reason for reduced accuracy in sentences with many phrases is the presence of transition signals. The more words in each sentence practiced, the higher the likelihood of numerous transition signals being created, which is highly relevant to time. Each signal has a different time delay, and as a result, the system developed has not yet demonstrated strong detection capabilities in the presence of time delays. Figure 10 is the following illustration correlates words in sentences with time. The "Tr" code in between words serves as a label for transitions represented by the character "$" in the dataset.

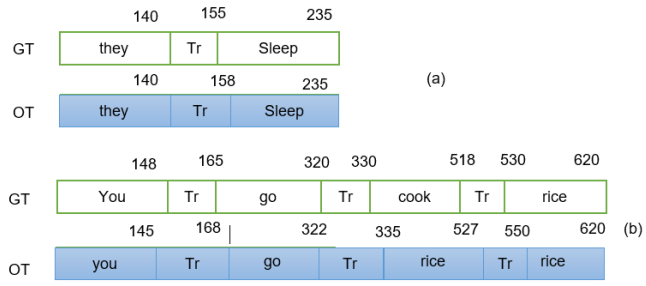


Fig. 10 Segmentation of sentences into words in sign language

Based on the illustration in Figure 10, we show the difference in accuracy between two words and four words in a sentence. The accuracy obtained in two words is better than that obtained in four words because they are closely related to time. (t). The duration of the training data and the real-time testing time are very influential. The duration of the training data should be the same as the real-time testing time, but the duration generated by each word will be different if the number of word transitions increases. The duration of the inter-word transition is indefinite, depending on the transition signal performed by the deaf. In Figure 10 (b) above, you can clearly see the frame shift towards time after performing the four-word signal. GT is the frame time according to the training data, while OT is the output at the time of the test. If the GT frame time differs from the OT, then it is likely to cause a detection error. Therefore, the removal of transition signals is highly recommended to remove noise from word movements in sentence signals.

B. *Evaluations using Signed word.*

Here, we show how well the improved LSTM model recognizes single sign words. An average recognition rate of 85% from 20 different sign terms has been noted.

TABLE III
WORD DETECTION USING LSTM MODIFIED

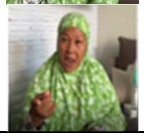
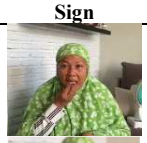
NO	Sign	Word	Detected
1		Love	Love
2		There	There
3		You	You
4		See	Two
5		Cry	Cry
6		Everything	Ever
7		Remember	Remember
8		Hear	Hear
9		Laugh	Laugh
10		Sleep	Sleep
11		Ball	We
12		They	There
13		Play	Play
14		We	We
15		Cook	Cook
16		Rice	Rice
17		Market	Market
18		Go	Go
19		Honestly	Honestly
20		$	$

Table III shows data that there were 3 words that were incorrectly detected out of 20 words that were tested on 3 deaf people. This detection error can be influenced by training data or testing data that does not comply with the rules. Based on a direct test by the author against a fool then there are some gestures that are difficult to detect by the system and have less or less probability of detection. This is because the gestures of the vocabulary have similarities with the gestures of the other vocabularies, positions that are not identical to the positions at the time of the training data. Besides, what can cause a misdetection is an error at the time of taking the training data or even a difference in the light intensity when taking training data with the test data.

C. Comparative Analysis

The research conducted by Putra et al. [26] combines Mediapipe as a feature extraction tool and the Long Short-Term Memory (LSTM) method for training and validating the model. In this research, the input data consists of videos extracted using MediaPipe. To evaluate the performance of the LSTM model, the study employs a confusion matrix as one of its methods. This research indicates that the LSTM model achieved the highest accuracy rate of 82% in recognizing individual words within gestures, while the accuracy rate in sentence recognition reached 48%. The training process for the LSTM model required a total time of 10 minutes and 50 seconds.

In another study by Agrawal et al. [32], a folder was created for each of the ten moves, and every 80 subfolder moves were created. The subfolder can be considered a video folder; within each subfolder, there are 30 frames, each in the form of a NumPy array containing landmark values that are detected and extracted using Mediapipe Holistic Solution. LSTM networks are trained to use data and provide 90% accuracy on test data. Finally, the system is tested using real-time data that is directly inserted into the model, and the results are displayed on the screen for each movement. Some delays were observed when recognizing movements in real-time: the test per word reached 65%, while the test per sentence reached only 43% accuracy.

The same research was conducted in this study, where real-time video input was used, employing Mediapipe as the feature extraction tool. However, the difference here lies in the modified LSTM method used. The modification aimed to reduce detection errors in sentences involving transitions between words. The results of this research, using the modified LSTM, demonstrated superior performance compared to the previous study. Accuracy was measured using confusion matrices and direct testing, revealing an accuracy of 80% in sentence detection and 85% in word detection. The training time for the data was reduced to just 9 minutes and 45 seconds, respectively, according to the presentation of Table IV on identification accuracy.

TABEL IV
COMPARATIVE LSTM AND MODIFIED LSTM ACCURACY

Author	Model	Sign Word Recognition	Sign Sentence Recognition	Training Time
Putra et al	LSTM	82%	48%	650 s
Agrawal et al.	LSTM	65%	45%	-
Ridwang et al.	Modified LSTM	85%	80%	585 s

This research makes a valuable contribution to the Human Action Recognition (HAR) field, showcasing the potential of utilizing MediaPipe and modified LSTM for feature extraction and model training. The obtained accuracy levels, particularly in word recognition, underscore the success of this method in addressing HAR challenges.

IV. CONCLUSION

In this study, we have developed a novel framework for sustainable SLR employing Mediapipe and cameras. The addition of sign and sentence words has also been suggested for the modified LSTM architecture. A data set of 20 distinct sign words was utilized to train the model. Our assessment of the technique is based on 320 signed sentences written by four signatories. On signed sentences and isolated marked words, average accuracy levels of 80.0% and 85% have been noted. In the future, additional training data for greater model learning can enhance introduction performance and get a new method for the detection transition sign.

ACKNOWLEDGMENT

The authors thank the Ministry of Education, Culture, Research and Technology of the Republic Indonesia, Universitas Hasanuddin and Universitas Muhammadiyah Makassar for provision of the research facilities.

REFERENCES

[1] B. Sundar and T. Bagyammal, "American Sign Language Recognition for Alphabets Using MediaPipe and LSTM," *Procedia Comput Sci*, vol. 215, pp. 642–651, 2022, doi:10.1016/j.procs.2022.12.066.

[2] Y. SHI, Y. LI, X. FU, M. I. A. O. Kaibin, and M. I. A. O. Qiguang, "Review of dynamic gesture recognition," *Virtual Reality and Intelligent Hardware*, vol. 3, no. 3. KeAi Communications Co., pp. 183–206, Jun. 01, 2021. doi:10.1016/j.vrih.2021.05.001.

[3] D. K. Jain, A. Kumar, and S. R. Sangwan, "TANA: The amalgam neural architecture for sarcasm detection in indian indigenous language combining LSTM and SVM with word-emoji embeddings," *Pattern Recognit Lett*, vol. 160, pp. 11–18, Aug. 2022, doi:10.1016/j.patrec.2022.05.026.

[4] Y. S. Tan, K. M. Lim, and C. P. Lee, "Hand gesture recognition via enhanced densely connected convolutional neural network," *Expert Syst Appl*, vol. 175, Aug. 2021, doi:10.1016/j.eswa.2021.114797.

[5] P. K. Athira, C. J. Sruthi, and A. Lijiya, "A Signer Independent Sign Language Recognition with Co-articulation Elimination from Live Videos: An Indian Scenario," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 771–781, Mar. 2022, doi:10.1016/j.jksuci.2019.05.002.

[6] R. O. Maimon-Mor *et al.*, "Talking with Your (Artificial) Hands: Communicative Hand Gestures as an Implicit Measure of Embodiment," *iScience*, vol. 23, no. 11, Nov. 2020, doi:10.1016/j.isci.2020.101650.

[7] V. Adithya and R. Rajesh, "Hand gestures for emergency situations: A video dataset based on words from Indian sign language," *Data Brief*, vol. 31, Aug. 2020, doi: 10.1016/j.dib.2020.106016.

[8] A. P. G and A. P. k, "Design of an integrated learning approach to assist real-time deaf application using voice recognition system," *Computers and Electrical Engineering*, vol. 102, p. 108145, Sep. 2022, doi:10.1016/j.compeleceng.2022.108145.

[9] C. Hinchcliffe *et al.*, "Language comprehension in the social brain: Electrophysiological brain signals of social presence effects during syntactic and semantic sentence processing," *Cortex*, vol. 130, pp. 413–425, Sep. 2020, doi:10.1016/j.cortex.2020.03.029.

[10] M. Suneetha, P. MVD, and K. PVV, "Multi-view motion modelled deep attention networks (M2DA-Net) for video based sign language recognition," *J Vis Commun Image Represent*, vol. 78, p. 103161, Jul. 2021, doi:10.1016/J.JVCIR.2021.103161.

[11] K. Sadeddine, Z. F. Chelali, R. Djeradi, A. Djeradi, and S. Ben Abderrahmane, "Recognition of user-dependent and independent static hand gestures: Application to sign language," *J Vis Commun Image Represent*, vol. 79, p. 103193, Aug. 2021, doi:10.1016/j.jvcir.2021.103193.

[12] L. R. Cerna, E. E. Cardenas, D. G. Miranda, D. Menotti, and G. Camara-Chavez, "A multimodal LIBRAS-UFOP Brazilian sign language dataset of minimal pairs using a microsoft Kinect sensor," *Expert Syst Appl*, vol. 167, p. 114179, Apr. 2021, doi:10.1016/j.eswa.2020.114179.

[13] R. Solgi, H. A. Loáiciga, and M. Kram, "Long short-term memory neural network (LSTM-NN) for aquifer level time series forecasting using in-situ piezometric observations," *J Hydrol (Amst)*, vol. 601, Oct. 2021, doi:10.1016/j.jhydrol.2021.126800.

[14] L. Gao, H. Li, Z. Liu, Z. Liu, L. Wan, and W. Feng, "RNN-Transducer based Chinese Sign Language Recognition," *Neurocomputing*, vol. 434, pp. 45–54, Apr. 2021, doi:10.1016/j.neucom.2020.12.006.

[15] S. Subburaj and S. Murugavalli, "Survey on sign language recognition in context of vision-based and deep learning," *Measurement: Sensors*, vol. 23, p. 100385, Oct. 2022, doi:10.1016/j.measen.2022.100385.

[16] K. Anand, S. Urolagin, and R. K. Mishra, "How does hand gestures in videos impact social media engagement - Insights based on deep learning," *International Journal of Information Management Data Insights*, vol. 1, no. 2, Nov. 2021, doi:10.1016/j.jjimei.2021.100036.

[17] R. Solgi, H. A. Loáiciga, and M. Kram, "Long short-term memory neural network (LSTM-NN) for aquifer level time series forecasting using in-situ piezometric observations," *J Hydrol (Amst)*, vol. 601, Oct. 2021, doi:10.1016/j.jhydrol.2021.126800.

[18] P. K. Athira, C. J. Sruthi, and A. Lijiya, "A Signer Independent Sign Language Recognition with Co-articulation Elimination from Live Videos: An Indian Scenario," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 771–781, Mar. 2022, doi:10.1016/j.jksuci.2019.05.002.

[19] G. A. Rao and P. V. V. Kishore, "Selfie video based continuous Indian sign language recognition system," *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 1929–1939, Dec. 2019, doi:10.1016/j.asej.2016.10.013.

[20] S. K. Devi and S. CN, "Intelligent Deep Learning Empowered Text Detection Model from Natural Scene Images," *Int J Adv Sci Eng Inf Technol*, vol. 12, no. 3, pp. 1263–1268, 2022, Accessed: Nov. 07, 2023. [Online]. doi:10.18517/ijaseit.12.3.15771.

[21] R. Sreemathy, M. P. Turuk, S. Chaudhary, K. Lavate, A. Ushire, and S. Khurana, "Continuous word level sign language recognition using an expert system based on machine learning," *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 170–178, Jun. 2023, doi:10.1016/J.IJCCE.2023.04.002.

[22] M. A. Almasre and H. Al-Nuaim, "A comparison of Arabic sign language dynamic gesture recognition models," *Heliyon*, vol. 6, no. 3, p. e03554, Mar. 2020, doi: 10.1016/j.heliyon.2020.E03554.

[23] Y. S. Tan, K. M. Lim, and C. P. Lee, "Hand gesture recognition via enhanced densely connected convolutional neural network," *Expert Syst Appl*, vol. 175, Aug. 2021, doi:10.1016/j.eswa.2021.114797.

[24] R. Gupta and A. Kumar, "Indian sign language recognition using wearable sensors and multi-label classification," *Computers & Electrical Engineering*, vol. 90, p. 106898, Mar. 2021, doi:10.1016/j.compeleceng.2020.106898.

[25] J. Bora, S. Dehingia, A. Boruah, A. A. Chetia, and D. Gogoi, "Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning," *Procedia Comput Sci*, vol. 218, pp. 1384–1393, Jan. 2023, doi:10.1016/j.procs.2023.01.117.

[26] I. A. Putra, O. D. Nurhayati, and D. Eridani, "Human Action Recognition (HAR) Classification Using MediaPipe and Long Short-Term Memory (LSTM)," *Teknik*, vol. 43, no. 2, pp. 190–201, Aug. 2022, doi:10.14710/teknik.v43i2.46439.

[27] B. Subramanian, B. Olimov, S. M. Naik, S. Kim, K. H. Park, and J. Kim, "An integrated mediapipe-optimized GRU. model for Indian sign language recognition," *Sci Rep*, vol. 12, no. 1, Dec. 2022, doi:10.1038/s41598-022-15998-7.

[28] R. Ridwang, I. Nurtanio, A. Ahmad Ilham, and S. Syafaruddin, "Deaf Sign Language Translation System with Pose and Hand Gesture Detection Under LSTM-Sequence Classification Model," *ICIC Express Letters*, vol. 17, no. 7, pp. 809–816, 2023, doi:10.24507/icicel.17.07.809.

[29] Q. Xiao, X. Chang, X. Zhang, and X. Liu, "Multi-Information Spatial-Temporal LSTM Fusion Continuous Sign Language Neural Machine Translation," *IEEE Access*, vol. 8, pp. 216718–216728, 2020, doi:10.1109/access.2020.3039539.

[30] B. Verma, "A two stream convolutional neural network with bi-directional GRU. model to classify dynamic hand gesture," *J Vis Commun Image Represent*, vol. 87, p. 103554, Aug. 2022, doi:10.1016/j.jvcir.2022.103554.

[31] Q. Xiao, X. Chang, X. Zhang, and X. Liu, "Multi-Information Spatial-Temporal LSTM Fusion Continuous Sign Language Neural Machine Translation," *IEEE Access*, vol. 8, pp. 216718–216728, 2020, doi: 10.1109/access.2020.3039539.

[32] A. S. Agrawal, A. Chakraborty, and C. M. Rajalakshmi, "Real-Time Hand Gesture Recognition System Using MediaPipe and LSTM," *International Journal of Research Publication and Reviews*, vol. 3, pp. 2509–2515, 2022.