Detecting signal transition in dynamic sign language using the R-GB LSTM method

Ridwang [a,1,*], Adriani [b,2], Rahmania [b,3], Mus'ab Sahrim [b,4], Asep Indra Syahyadi [c,5], Haris Setiaji [d,6]

[a] Universitas Muhammadiyah Makassar, Jl. Sultan Alauddin No.259, Makassar 90221, Indonesia
[b] Universiti Sains Islam Malaysia (USIM), Bandar Baru Nilai, 71800 Nilai Negeri Sembilan, Malaysia
[c] Universiti Islam Negeri Alauddin Makassar, Jl. Sultan Alauddin No.63, Gowa 92113, Indonesia
[d] Institut Agama Islam Negeri Metro Lampung, Jl. Ki Hajar Dewantara No.15A, Kota Metro 34112 Indonesia
[1] ridwang@unismuh.ac.id; [2] adriani@unismuh.ac.id; [3] rahmania.rahmania@unismuh.ac.id; [4] musab@usim.edu.my; [5] asep@uin-alauddin.ac.id;
[6] harissetiaji@metrouniv.ac.id
* corresponding author

ARTICLE INFO

Article history
Received December 9, 2023
Revised March 26, 2024
Accepted April 7, 2024
Available online May 31, 2024

Keywords
Deaf people
R-GB LSTM
Word sign
Sentence sign
Sign language

ABSTRACT

Sign Language Recognition (SLR) helps deaf people communicate with normal people. However, SLR still has difficulty detecting dynamic movements of connected sign language, which reduces the accuracy of detection. This results from a sentence's usage of transitional gestures between words. Several researchers have tried to solve the problem of transition gestures in dynamic sign language, but none have been able to produce an accurate solution. The R-GB LSTM method detects transition gestures within a sentence based on labelled words and transition gestures stored in a model. If a gesture to be processed during training matches a transition gesture stored in the pre-training process and its probability value is greater than 0.5, it is categorized as a transition gesture. Subsequently, the detected gestures are eliminated according to the gesture's time value (t). To evaluate the effectiveness of the proposed method, we conducted an experiment using 20 words in Indonesian Sign Language (SIBI). Twenty representative words were selected for modelling using our R-GB LSTM technique. The results are promising, with an average accuracy of 80% for gesture sentences and an even more impressive accuracy rate of 88.57% for gesture words. We used a confusion matrix to calculate accuracy, specificity, and sensitivity. This study marks a significant leap forward in developing sustainable sign language recognition systems with improved accuracy and practicality. This advancement holds great promise for enhancing communication and accessibility for deaf and hard-of-hearing communities.

1. Introduction

Sign language is the primary language for many deaf people. It is not merely an alternative form of spoken language but a fully-fledged language with its own grammar, syntax, and vocabulary [1]. As such, recognizing and understanding sign language is crucial for facilitating effective communication with deaf individuals. SLR technology bridges the communication gap between deaf and hearing people. By converting sign language into a form that machines or electronic devices can understand, such as text or speech, SLR allows deaf people to communicate with those who do not know sign language. This is especially important in situations where a sign language interpreter is not available, such as when communicating with people who are not proficient in sign language or when using technology such as telephones or computers [2]. SLR also facilitates the development of applications and technologies that

can assist deaf people in their daily lives [3]. This includes real-time sign language translation, more sophisticated hearing aids, and alternative communication systems that are more efficient. In this way, SLR not only facilitates communication between deaf and hearing individuals but also creates opportunities for innovation to enhance the quality of life and social inclusion of the deaf community [4].

While deep learning techniques have emerged as a powerful tool for continuous sign language recognition (SLR), challenges like segmentation, coarticulation, and context dependence in dynamic signs persist. Alaghband M etc highlights the potential of CNN and RNN to address these complexities by capturing both spatial and temporal features of sign language [5]. Research has since delved deeper into these possibilities. According to Strobel G cs, 3D CNN architectures use depth camera data to capture the complex 3D structure of hand shapes. This makes recognition more accurate than with traditional 2D methods. Furthermore, sensor fusion, which combines data from RGB cameras (capturing colour and appearance) and depth sensors (capturing 3D information), provides a richer representation of signs. This, as shown in a separate study, allows SLR systems to account for both hand and body posture, which is crucial for understanding context in dynamic sign language [6].

Beyond improved feature extraction, end-to-end learning offers a simplified development process. Deep learning models can automatically learn features and perform recognition, as explored in another study. This approach holds promise for dynamic SLR tasks [7]. Looking toward the future, research is expanding beyond recognition. Zeyu Liang et.al introduces an attention-based encoder-decoder framework for sign language translation, aiming to improve the fluency and naturalness of translating signed utterances into spoken language or text [8]. The field of dynamic SLR is rapidly evolving. A 2023 survey provides a comprehensive overview of the latest advancements and potential applications, including real-time communication, education, and assistive technologies [9]. Future research directions like personalization, sign language translation advancements, and open-set recognition for handling unseen signs offer exciting possibilities for further bridging the communication gap. In essence, deep learning has revolutionized the field of dynamic SLR, paving the way for a future where sign language can be seamlessly understood and translated, fostering greater inclusivity and communication for the deaf community [10].

The research entitled "Movement Epenthesis Detection for Continuous Sign Language Recognition" [11] was conducted in 2017 to detect and eliminate Epenthesis movements, which are additional movements in a sign language sequence. The detection method used ME detection, which symbolises its frames with H_code. If the H_code value is < T1, then the epenthesis movement is deleted or categorized. If the H_code value is > T1, then it is categorized as a gesture. The T1 value is a threshold value taken from the minimum height of the square area boundary. The accuracy achieved in the per-word test was 92.8%, while the per-sentence test result was 78%. Research conducted by Boris Mocialov in 2020 on sign language recognition (SLR) used the deep learning method to detect gestures from video data and overcome the influence of epenthesis in a sentence [12]. This research used a dataset consisting of 474 images taken using five different types of videos. The images were then analyzed using the deep learning method, namely the convolutional neural network, to predict image quality and identify important features that affect image quality, such as brightness level, contrast, and image detail, so that it could distinguish between word features and transitions. This CNN method is used to perform classification and produce outputs with an accuracy of 80% [13].

None of the aforementioned studies demonstrate a technique for identifying and eliminating transition signals in continuous sign languages, which could enhance the accuracy of dynamic sign language detection. Therefore, this study's main contribution lies in two key aspects. Firstly, it introduces a unique technique for dynamic Indonesian sign language that recognizes and removes transition signals. Secondly, it demonstrates that the accuracy of dynamic gesture detection can be enhanced by removing transition gestures, which often introduce noise into the sign language detection system, utilizing R-GB LSTM.

This paper aims to explain the R-GB LSTM method, which is a combination of GRU, Backpropagation, and LSTM, used to detect and eliminate transition gestures considered noise and increase accuracy.

2. Method

This section describes our neural network design for continuous SLR using Camera and Mediapipe Library, which is based on R-GB LSTM. The framework's flow diagram is shown in Fig. 1, where the sign inputs are acquired via the Camera.

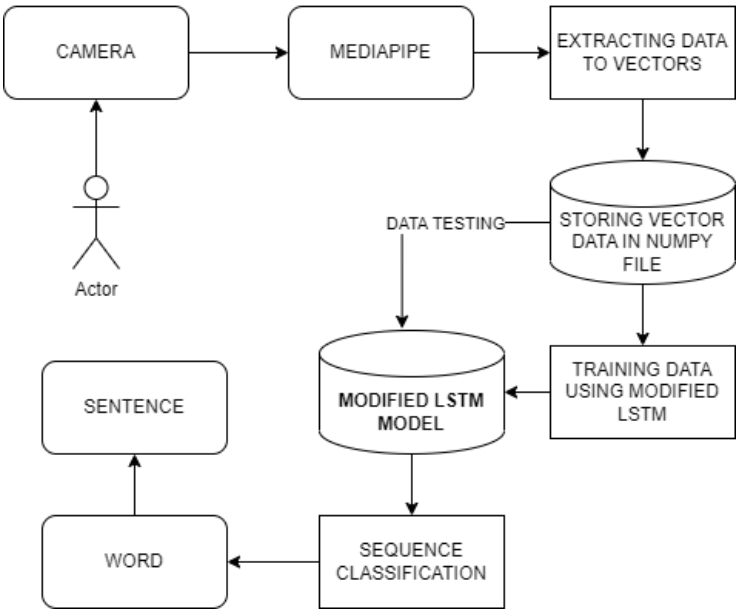


Fig. 1. R-GB LSTM Diagram

2.1. Pre-processing

In the pre-processing pipeline employing the Mediapipe library, five crucial steps are involved in extracting features from photos and adjusting them based on camera detection findings. Initially, we transform videos into frames, generating 30 frames for each video loop. Following this, images undergo a conversion from the BGR to RGB colour space using OpenCV functions, facilitating subsequent processing steps [14]. Pose detection is then conducted utilizing Mediapipe's BlazePose detectors and subsequent landmark models. This process extracts three distinct regions of interest (ROIs) for two hands and the face. In instances where the accuracy of the pose model falls short, a cropping model intervenes, applying spatial transformations to rectify erroneous hand ROIs. Remarkably, this correction process consumes merely 10% of the hand model's inference time [15].

2.2. Feature Extraction

The feature extraction process involves obtaining 33 landmarks or key points from poses, with each pose represented in three dimensions (x, y, and z), resulting in a total of 132 data points per pose [6], [16]. For hand feature extraction, three variables are multiplied by 21 key points, yielding 63 data points for each hand, thus totalling 126 data points for both hands. Additionally, features are extracted from the face, resulting in 468 key points in three dimensions, totalling 1404 data points [17]. These data points are then flattened using TensorFlow's flat function to ensure one-dimensional output. Subsequently, they are concatenated using Python's String Concatenate function to merge hand and pose key points. This comprehensive set comprises 1662 key points from poses, hands, and faces, which serve as input for subsequent processing in the system architecture, particularly for training the LSTM network [18].

2.3. R-GB LSTM Network

Fig. 2 shows the architecture R-GB LSTM method which is developed from the LSTM method with an initial architecture using 3 gates: forget gate, input gate, and output gate. R-GB LSTM stands for Ridwang-GRU Backpropagation LSTM, where this method combines LSTM, GRU, and backpropagation techniques.

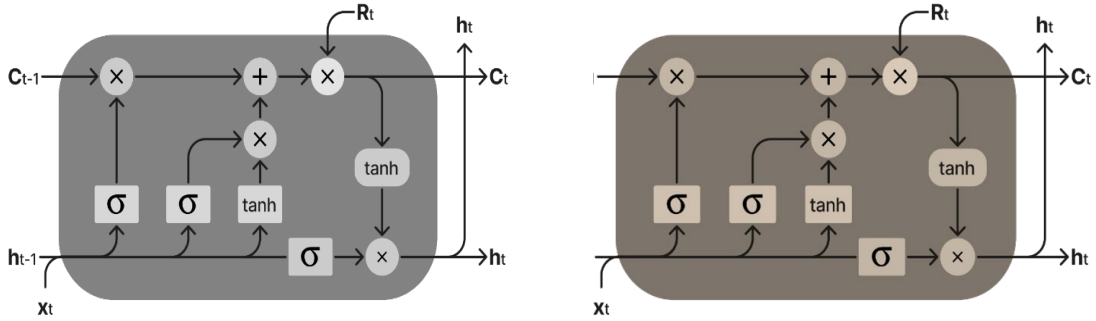


Fig. 2. Architecture R-GB LSTM

In addition to the sigmoid activation function, the input and output gates additionally contain a tanh activation function. The sigmoid activation function is used to produce output values of the gates ranging from 0 to 1, while the tanh activation function produces values ranging between -1 and 1. Several variables are used, including Ct, which represents the Cell State or long-term memory [19]. This memory carries information from the beginning to the end, allowing LSTM to retain long-term information and is suitable for prediction processes requiring long dependencies. In addition to having the Ct memory, LSTM also has the ht memory, commonly known as the hidden state. The hidden state is a short-term memory used as input data in the next layer and also as the output of the LSTM unit [20]. Furthermore, another variable used is the input variable (xt), representing the input variable at each time step in LSTM. Based on this data, the formulas for several processes of LSTM can be observed in the formulas below [12], [21].

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

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

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

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

$$\boldsymbol{C_{t_{new}} = C_t * R_t} \tag{5}$$

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

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

2.4. Model Training

Our model consists of three layers, which are a dense layer, a layer of R-GB LSTM units with RELU activation function, and a Softmax output layer for multi-category classification [22]. We utilized the Adadelta optimizer, which is renowned for its ability to withstand noisy gradients and eliminate the requirement for manual learning rate modifications. To efficiently model the framework, we first train the model using discrete sign motions and then refine it with continuous gestures. Introducing a variable-length transition between gestures or states converts discrete motions into continuous signs [23].

The softmax function, illustrated in Fig. 3, assigns probabilities to each potential class, ensuring they all sum up to 1. These probabilities are derived from the output vector's values. Each value in the output vector is then converted into its corresponding class label [24]. It's important to note that softmax provides a probability value for every possible class, even if a class label has multiple words. Finally, in common practice, these softmax outputs are converted to their corresponding class labels and stored as part of the model [25][25].

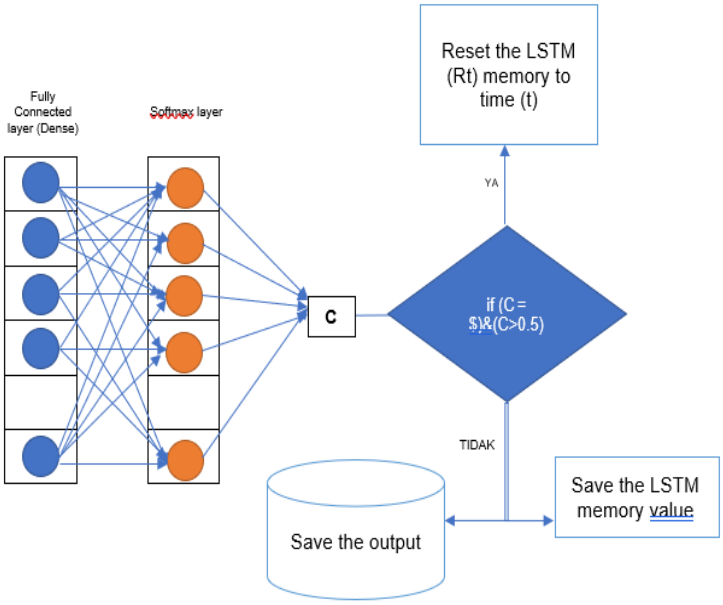


Fig. 3. Modification diagram of output Softmax

2.5. Model Validation

Every experiment adhered to a set protocol to assess the model's performance. First, training (80%) and validation (20%) sets of data were separated. The training data was then used to train the model. Each experiment recorded two important metrics during the evaluation of the validation set: total accuracy and a confusion matrix with data for each of the twenty classes [26]. After that, a more intricate table (Table 2) was created from this larger confusion matrix, which displayed the quantity of TP, TN, FP, and FN for each class. The model's total ability to accurately detect cases is reflected in its accuracy. Specificity gauges the model's capacity to accurately categorize real negatives, whereas sensitivity concentrates on how well the model detects true positives [27]. All of these measurements come from the connections between TP, TN, FP, and FN [28].

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

$$Se = \frac{TP}{TP+FN} \tag{9}$$

$$Sp = \frac{TN}{TN+FP} \tag{10}$$

The TP, TN, FP, and FN can be used to generate metrics such as the MCC, FM, and BM to show the statistical relevance of each class. The MCC measure evaluates the degree of agreement between expected and observed binary classifications. The degree to which observed and predicted binary classifications agree with one another is assessed using FM. The chance that a choice will be made with information is determined using Bayesian modelling BM [29].

$$MCC = \frac{(TP\ x\ TN)-(FP\ x\ FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{11}$$

$$FM = \sqrt{\frac{TP}{TP+FP} x \frac{TP}{TP+FN}} \quad (12)$$

$$BM = Se + Sp - 1 \quad (13)$$

3. Results and Discussion

The study focused on training 82 words, selected according to their word type: 15 nouns, 27 verbs, 23 adjectives, 9 adverbs, and 8 question words. To focus this study, 19 single sign words were taken, plus one transitional sign marked with the symbol '&' originating from four participants who expressed their interest in participating in this sign language data collection. Each signer uttered each sign word at least thirty times. The result is that 2,280 sign words (19 * 30 * 4) were recorded. Each sign word is described in Table 1. In a sign sentence, the '$' sign indicates the transitional movement between two consecutive letters

Table 1. Accessible sign language within the dataset

Love	Everything	Wall	Noon
Where	Remember	They	Market
You	Miss	Play	Go
See	Laugh	We	Honestly
Think	Run	Buy	$

.

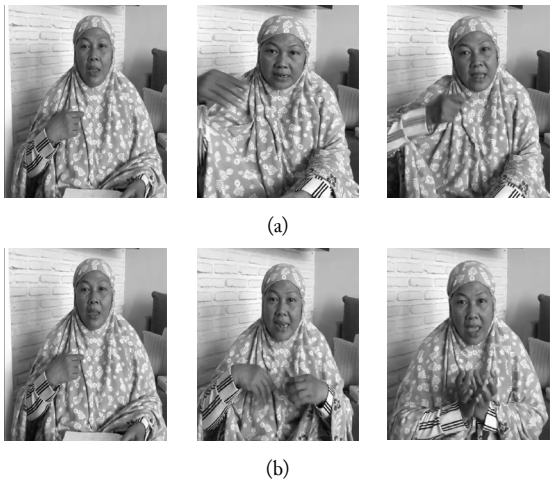
In this study, to calculate accuracy, sensitivity, and specificity, a confusion matrix containing 20 classes was generated for each of the five experiments. This matrix was then converted into matrices that represented TP, TN, FP, and FN. To perform a comprehensive examination of the findings, averages across five iterations were tabulated for each word for ACC, Se, and Sp, as illustrated in Table 2.

Table 2. Accuracy(ACC), Sensitivity (Se) and Specificity (Sp), MCC, FM and BM

Class	Acc	S_p	S_e	MCC	FM	BM
Love	100	100	100	100	100	100
Where	100	100	100	100	100	100
You	99	100	89	94	94	89
See	99	75	100	86	87	99
Think	99	100	75	86	87	75
Everything	98	80	100	89	89	98
Remember	99	86	100	92	93	99
Miss	99	83	100	91	91	99
Laugh	100	100	100	100	100	100
Run	100	100	100	100	100	100
Wall	100	100	100	100	100	100
They	99	100	75	86	87	75
Play	99	100	80	89	89	80
We	100	100	100	100	100	100
Buy	100	100	100	100	100	100
Noon	100	100	100	100	100	100
Market	99	80	100	89	89	99
Go	100	100	100	100	100	100
Honestly	99	100	86	92	93	86
$	100	100	100	100	100	100

Table 2 shows the average ACC, Se, Sp, MCC, FM, and BM using the R-GB LSTM method. The observation results show that all data almost reach the highest value. The accuracy of the word

"everything" has the lowest accuracy, which is 98. The lowest Specificity value is 75 for the word "See" and the lowest Sensitivity is for the words "think" and "they". The reason for this is that the gestures of the two terms bear some resemblance to each other. Additionally, the metrics MCC, FM, and BM are influenced by the presence of both high and low values of FN and FP.



(a)

(b)

Fig. 4. Example Sentence sign(a)we go market (b)we play ball

The Fig. 4 shows signs for a sentence. In the first picture (a), it starts with "Go" signed like throwing something, then "Market" with a hand holding money, and finally "We" with a pointing finger. In the second picture (b), it starts with "We" again (pointing finger), then "Play" with two hands moving together, and ends with "Ball" by showing two hands holding a ball.

3.1. Recognition of Signed Sentences

We trained the suggested R-GB LSTM model on signed words to show that it can recognize sentences. Using a RESET LSTM state condition for transition gestures ($), the network was trained across 150 epochs in 5 minutes on an NVIDIA Intel GPU workstation. The learning curve of the model is shown in Fig. 5, wherein training and validation errors show a decrease and saturation [30].

The graph shows Fig 5 that the training accuracy (solid line) increases as the number of epochs increases. This means that the model is performing better on the training data as it sees more of it. The testing accuracy (dashed line) also increases initially, but it starts to plateau around 100 epochs. This suggests that the model may be overfitting the training data. Therefore, the number of epochs chosen was up to 150 to prevent overfitting, and the model's accuracy has also increased. As the number of epochs increases, the graph indicates a decrease in training loss (solid line), signifying that the model is learning from the training data and enhancing its performance. Initially, the testing loss (dashed line) also decreases, but it begins to level off around 120 epochs.

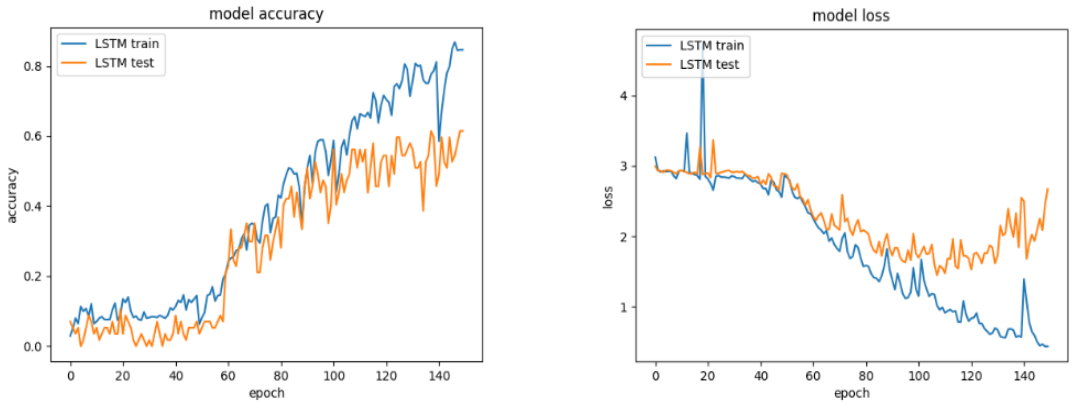


Fig. 5. Accuracy and Loss Model on Epoch 150

The evaluation showed that the correctness of signed sentences was 80.0% on average. Phrases consisting of six sign words had the lowest accuracy of 77%, while sentences with two words recorded the highest recognition rate of 84.0%. The variation in sign identification accuracy for sentences of different durations is displayed in Fig. 6.

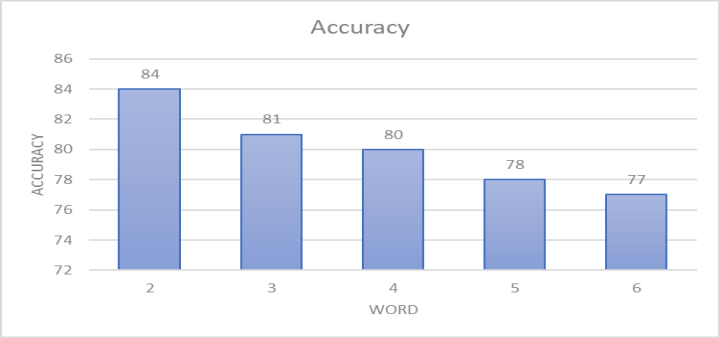


Fig. 6. Accuracy of recognition based on signed sentence length

3.2. Sign word

We demonstrate the proficiency of the enhanced R-GB LSTM model in the recognition of individual sign words, achieving an impressive average recognition rate of 88.57% across 19 distinct sign terms. Fig. 7 presents the complete confusion matrix for all sign words visually, giving a thorough picture of the model's effectiveness in differentiating between sign terms.

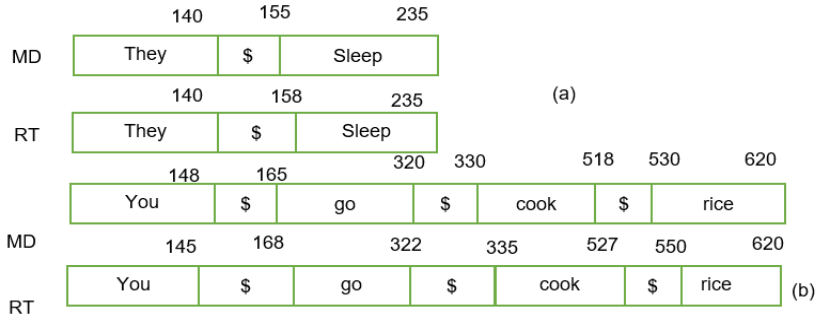


Fig. 7. Word division in sign language sentences

Fig. 7 provides an example of sentence segmentation into words according to the data in the dataset. Fig. 7 shows a sentence that has been tested using the R-GB LSTM method, so the model and dataset used already have transition gestures in them. In Figure (a), the sentence "mereka Tidur" is shown, which consists of 3 gestures: 2-word gestures and 1 transition gesture. The sentence detection using two words resulted in 100% accuracy. This is because the frame position between the actual state and the output is almost the same, although there is a slight difference at the beginning of the transition frame and the second word frame. In Figure (b), there is a significant difference between the initial frames of the last word, so the probability of misdetection is quite high. This serves as a lesson for future research to maximize the data collection method or dataset so that internal and external transition gestures can be minimized, thus improving the accuracy of sign language translation.

3.3. Comparative Analisis

In this section, we conducted an indirect comparison between our proposed Sign Language Recognition (SLR) framework and several state-of-the-art methodologies. Kong et al. [9] introduced a continuous SLR system for American Sign Language (ASL) using a cyber glove, employing a conditional random field (CRF) model with two layers for recognition. Their approach involved segmenting before recognizing continuous signed texts, employing a vocabulary of 107 signs in 74 signed sentences. Similarly, in [31], the authors developed a continuous SLR system for Arabic Sign Language (ArSL) based on image/video processing. Their method incorporated discrete cosine transform (DCT)-based

characteristics and utilized an HMM classifier for recognition. With a sign vocabulary of 80 symbols and 40 sign sentences, their methodology reported a recognition performance of 72.3%. In contrast, our dataset comprises 157 unique signed sentences, showcasing the comprehensive nature of our evaluation and resulting in a notable recognition performance [32].

Additionally, we performed a preliminary accuracy comparison between the standard LSTM model and the suggested alterations to the LSTM architecture. 256 hidden states, 256 group sizes, and adaptive learning levels ranging from 0.001—which decrease by a factor of 0.5 every 20 epochs—were used to train the conventional LSTM [33]. Both signed words and signed sentences were taken into consideration separately in this evaluation. The findings shown in Table 3 indicate that our proposed models outperformed the baseline LSTM architecture in terms of performance. Notably, they achieved higher recognition accuracy of signed words and phrases than the standard LSTM by a margin of 5.07% and 26%, respectively.

Table 3. Compare LSTM and R-GB LSTM Result

Model	Sign Word Recognition	Sign Sentence Recognition
LSTM	83.50%	54%
R-GB LSTM	88.57%	80%

The proposed method for recognizing transition and elimination gestures is R-GB LSTM because it can perform well according to the available training data. For further research, more pre-training data can be developed so that the gesture detection capability is more accurate and the computation time used will also be faster.

4. Conclusion

The proposed research has developed an innovative model for detecting transition gestures in deaf sign language translation systems. The spatial-temporal properties of a gesture sequence can be extracted using R-GB LSTM, which is particularly useful for dynamic gesture recognition. For real-time application usage, integrating the R-GB LSTM and sequence classification model can eliminate noise within the gesture sequence, making the model's work easier and improving the accuracy result. Based on the research results, the proposed method's accuracy reached 80% for sentence testing. Consequently, the proposed method is suitable for identifying dynamic transition gestures in deaf sign language translation systems. For future research, it is recommended to develop a new strategy that can be used to modify the output layer of the method used to improve the accuracy of dynamic gesture detection from deaf sign language.

Acknowledgment

The author would like to express their gratitude to Universitas Muhammadiyah Makassar and the Internal Research Grant with Universiti Sains Islam Malaysia.

Declarations

Author contribution. All authors contributed equally. All authors read and approved the final paper
Funding statement. The research is supported by an internal research grant by the Mahamadiyah Makassar University with an international collaboration scheme.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] 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.

[2] A. A. Zare and S. H. Zahiri, "Recognition of a real-time signer-independent static Farsi sign language based on fourier coefficients amplitude," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 5, pp. 727–741, May 2018, doi: 10.1007/s13042-016-0602-3.

[3] M. Al-Qurishi, T. Khalid, and R. Souissi, "Deep Learning for Sign Language Recognition: Current Techniques, Benchmarks, and Open Issues," *IEEE Access*, vol. 9, pp. 126917–126951, 2021, doi: 10.1109/ACCESS.2021.3110912.

[4] A. Chaikaew, K. Somkuan, and T. Yuyen, "Thai Sign Language Recognition: an Application of Deep Neural Network," in *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, Mar. 2021, pp. 128–131, doi: 10.1109/ECTIDAMTNCON51128.2021.9425711.

[5] M. Alaghband, H. R. Maghroor, and I. Garibay, "A survey on sign language literature," *Mach. Learn. with Appl.*, vol. 14, p. 100504, Dec. 2023, doi: 10.1016/j.mlwa.2023.100504.

[6] A. Abbaskhah, H. Sedighi, and H. Marvi, "Infant cry classification by MFCC feature extraction with MLP and CNN structures," *Biomed. Signal Process. Control*, vol. 86, p. 105261, Sep. 2023, doi: 10.1016/j.bspc.2023.105261.

[7] 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," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 3, pp. 771–781, Mar. 2022, doi: 10.1016/j.jksuci.2019.05.002.

[8] Z. Liang, H. Li, and J. Chai, "Sign Language Translation: A Survey of Approaches and Techniques," *Electronics*, vol. 12, no. 12, p. 2678, Jun. 2023, doi: 10.3390/electronics12122678.

[9] S. Kausar and M. Y. Javed, "A Survey on Sign Language Recognition," in *2011 Frontiers of Information Technology*, Dec. 2011, pp. 95–98, doi: 10.1109/FIT.2011.25.

[10] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis," *Intell. Syst. with Appl.*, vol. 12, p. 200056, Nov. 2021, doi: 10.1016/j.iswa.2021.200056.

[11] A. Choudhury, A. Kumar Talukdar, M. Kamal Bhuyan, and K. Kumar Sarma, "Movement Epenthesis Detection for Continuous Sign Language Recognition," *J. Intell. Syst.*, vol. 26, no. 3, pp. 471–481, Jul. 2017, doi: 10.1515/jisys-2016-0009.

[12] B. Mocialov, G. Turner, K. Lohan, and H. Hastie, "Towards Continuous Sign Language Recognition with Deep Learning," *Proceeding Work. Creat. Mean. with Robot Assist. Gap Left by Smart Device*, p. 5, 2017, [Online]. Available at: https://homepages.inf.ed.ac.uk/hhastie2/pubs/humanoids.pdf.

[13] J. P. Sahoo, S. Ari, and S. K. Patra, "A user independent hand gesture recognition system using deep CNN feature fusion and machine learning technique," in *New Paradigms in Computational Modeling and Its Applications*, Elsevier, 2021, pp. 189–207, doi: 10.1016/B978-0-12-822133-4.00011-6.

[14] A. S. Agrawal, A. Chakraborty, and C. M. Rajalakshmi, "Real-Time Hand Gesture Recognition System Using MediaPipe and LSTM," *Int. J. Res. Publ. Rev.*, vol. 3, no. 4, pp. 2509–2515, 2022, [Online]. Available at: https://ijrpr.com/uploads/V3ISSUE4/IJRPR3693.pdf.

[15] 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.

[16] E. R. Swedia, A. B. Mutiara, M. Subali, and Ernastuti, "Deep Learning Long-Short Term Memory (LSTM) for Indonesian Speech Digit Recognition using LPC and MFCC Feature," in *2018 Third International Conference on Informatics and Computing (ICIC)*, Oct. 2018, pp. 1–5, doi: 10.1109/IAC.2018.8780566.

[17] T. S. Dias, J. J. A. Mendes, and S. F. Pichorim, "Comparison between handcraft feature extraction and methods based on Recurrent Neural Network models for gesture recognition by instrumented gloves: A case for Brazilian Sign Language Alphabet," *Biomed. Signal Process. Control*, vol. 80, p. 104201, Feb. 2023, doi: 10.1016/j.bspc.2022.104201.

[18] K. Anand, S. Urolagin, and R. K. Mishra, "How does hand gestures in videos impact social media engagement - Insights based on deep learning," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100036, Nov. 2021, doi: 10.1016/j.jjimei.2021.100036.

[19] A. A. Ilham, I. Nurtanio, Ridwang, and Syafaruddin, "Applying LSTM and GRU Methods to Recognize and Interpret Hand Gestures, Poses, and Face-Based Sign Language in Real Time," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 28, no. 2, pp. 265–272, Mar. 2024, doi: 10.20965/jaciii.2024.p0265.

[20] Ridwang, I. Nurtanio, A. A. Ilham, and Syafaruddin, "Deaf Sign Language Translation System With Pose and Hand Gesture Detection Under Lstm-Sequence Classification Model," *ICIC Express Lett.*, vol. 17, no. 7, pp. 809–816, 2023. [Online]. Available at: https://cir.nii.ac.jp/crid/1390296265971342976.

[21] S. C. Agrawal, A. S. Jalal, and C. Bhatnagar, "Recognition of Indian Sign Language using feature fusion," in *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI)*, Dec. 2012, pp. 1–5, doi: 10.1109/IHCI.2012.6481841.

[22] 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.

[23] Z. Zhang, C. Xu, J. Xie, Y. Zhang, P. Liu, and Z. Liu, "MFCC-LSTM framework for leak detection and leak size identification in gas-liquid two-phase flow pipelines based on acoustic emission," *Measurement*, vol. 219, p. 113238, Sep. 2023, doi: 10.1016/j.measurement.2023.113238.

[24] Y. Kartynnik, A. Ablavatski, I. Grishchenko, and M. Grundmann, "Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs," pp. 1-5, 2019. [Online]. Available at: https://static1.squarespace.com/static/5c3f69e1cc8fedbc039ea739/t/5d015ff133d4280001167610/1560371186690/6_CV4AR_Mesh.pdf.

[25] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta, "Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting," *Procedia Comput. Sci.*, vol. 135, pp. 89–98, Jan. 2018, doi: 10.1016/j.procs.2018.08.153.

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

[27] A. Mittal, P. Kumar, P. P. Roy, R. Balasubramanian, and B. B. Chaudhuri, "A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion," *IEEE Sens. J.*, vol. 19, no. 16, pp. 7056–7063, Aug. 2019, doi: 10.1109/JSEN.2019.2909837.

[28] M. A. As'ari, N. A. J. Sufri, and G. S. Qi, "Emergency sign language recognition from variant of convolutional neural network (CNN) and long short term memory (LSTM) models," *Int. J. Adv. Intell. Informatics*, vol. 10, no. 1, p. 64, Feb. 2024, doi: 10.26555/ijain.v10i1.1170.

[29] Ridwang, A. A. Ilham, I. Nurtanio, and - Syafaruddin, "Dynamic Sign Language Recognition Using Mediapipe Library and Modified LSTM Method," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 13, no. 6, pp. 2171–2180, Dec. 2023, doi: 10.18517/ijaseit.13.6.19401.

[30] C. Millar, N. Siddique, and E. Kerr, "LSTM Network Classification of Dexterous Individual Finger Movements," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 26, no. 2, pp. 113–124, Mar. 2022, doi: 10.20965/jaciii.2022.p0113.

[31] W. Abdul *et al.*, "Intelligent real-time Arabic sign language classification using attention-based inception and BiLSTM," *Comput. Electr. Eng.*, vol. 95, p. 107395, Oct. 2021, doi: 10.1016/j.compeleceng.2021.107395.

[32] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, "Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review," *Sensors*, vol. 20, no. 15, p. 4220, Jul. 2020, doi: 10.3390/s20154220.

[33] E. Pan, X. Mei, Q. Wang, Y. Ma, and J. Ma, "Spectral-spatial classification for hyperspectral image based on a single GRU," *Neurocomputing*, vol. 387, pp. 150–160, Apr. 2020, doi: 10.1016/j.neucom.2020.01.029.