Disambiguating Similar Sign Language Gestures: A Hybrid Approach Combining LSTM and ResNet50

Makoto Nakamura
Dept. of Sci. & Tech.
Gunma University
Gunma, Japan
t241b068@gunma-u.ac.jp

Dipanita Chakraborty
Division of Information Science
Nara Institute of Science and Technology
Nara, Japan
chakraborty.dipanita@naist.ac.jp

Kou Yamada
Dept. of Sci. & Tech.
Gunma University
Gunma, Japan
yamada@gunma-u.ac.jp

Kosin Chamnongthai
Dept. of ETE
KMUTT
Bangkok, Thailand
kosin.cha@kmutt.ac.th

***Abstract*—Sign Language Recognition systems often struggle with misclassification when encountering words with similar motion trajectories, a known limitation in methods relying solely on temporal features like LSTM networks. To address this problem, we hypothesize that distinct spatial features (hand shape) can disambiguate these confusing movements. We propose a hybrid architecture that combines a standard LSTM network, processing MediaPipe keypoint sequences to capture temporal dynamics, with a ResNet50 model dedicated to spatial feature extraction. The baseline "LSTM only" model achieves an overall Sensitivity of 91.52%. Our proposed "LSTM + ResNet50" hybrid model improves the overall Sensitivity to 95.15%. These results demonstrate that incorporating dedicated hand shape analysis can effectively mitigate errors caused by similar motion patterns. Future work will involve evaluating this architecture on larger datasets to further assess its generalization capabilities and compare accuracy.**

***Index Terms*—3D reconstruction, Sign Language Recognition, Deep Learning, Multi-View Classification**

I. Introduction

Sign Language is a non-verbal form of communication for hearing-impaired people. American Sign Language has the largest population of any language [1]. When hearing-impaired people use government services and medical care, accessibility issues arise due to communication gaps between hearing-impaired people and hearing people. Therefore, an automatic sign language interpretation system is needed. Various researchers conduct studies on automatic sign language interpretation systems.

According to papers such as [2], Abdullahi et al. address misclassification caused by similar hand movement trajectories in bimanual dynamic words. Furthermore, Chophuk et al. [3] address the problem that interpretation systems fail to recognize similar shapes, rotation, and movement (SRM) words. As a result, in both studies, the extracted hand features show high performance, and the accuracy of identifying each sign language word is also high. However, the discussions in these studies indicate that there are still challenges to overcome. Chophuk et al. point out that occlusion, where fingers overlap and hide each other, can cause a word (e.g., "keep") to be misclassified as another word (e.g., "sister"). Similarly, Abdullahi et al. analyze the existence of word clusters with low recognition scores and report that words with similar actions remain a "thorny" problem, prone to being misclassified as one another [2]. Long Short-Term Memory (LSTM) [4]/Bidirectional Long Short-Term Memory (BiLSTM) [2], [3]-based methods that use MediaPipe keypoints as input tend to rely too much on temporal "motion" features, failing to capture occlusions and subtle shape differences.

In this paper, to address the problem of misclassification that occurs when recognizing sign language words with similar movement trajectories, we propose the architecture, that combines LSTM and Deep Residual Learning for Image Recognition (Resnet50) [5], to disambiguate the problem. Specifically, we add LSTM, an action recognition model, to recognize hand movements and ResNet50, an image recognition model, to recognize hand shapes. This aims to resolve misclassification even for groups of words with similar actions by adding the subtle differences in hand shape learned by ResNet50 as discriminative information. In order to specifically verify the usefulness of adding this hand shape information, as a preliminary step, we select 11 words from Abdullahi et al.'s 40-word dataset and compare and evaluate the accuracy of word classification using LSTM with the accuracy of using our proposed LSTM + ResNet50.

II. Related Work

Abdullahi et al. [2] also use 3D sensors and address the problem of misclassification in double-hand dynamic sign words. They identify that the cause of misclassification is that many words possess similar hand motion trajectories at the beginning and end of the motion. To address this problem, they extract the "Hand Pause" feature-representing the preparation and retraction phases of the motion—which existing research overlooks. By inputting this feature into a multi-stacked deep BiLSTM network, they achieve a high accuracy of 97.98%. However, in the discussion, it is pointed out that words sharing many feature parameters (i.e., having very similar motions), such as "Please" and "Angry", remain a "thorny" problem as they are easily misclassified with each other.

Chophuk et al. [3] propose a "Backhand approach" that uses a 3D sensor mounted on the chest and tackles the recognition problem of the "SRM sign group" (words with similar Shape, Rotation, and Movement). They point out that conventional hand features struggle to distinguish similar words like

"brother" and "sister". As a solution, they introduce "spatial-temporal body parts and hand relationship (ST-BHR)" patterns as a new feature. By classifying these ST-BHR features with a two-layer BiLSTM, they achieve high recognition accuracy.

III. METHODOLOGY

In this section, we explain how to prepare the dataset, each model that makes up the architecture, and evaluate the test set and word classification.

A. Dataset preparation

We select 11 words ("again", "angry", "available", "bad", "bicycle", "big", "but", "car", "cheap", "clothes" and "cold") as recognition targets. As a dataset, we prepared 30 hand sign videos for each of these 11 words.

B. LSTM model for learning sign language movement

Next, we construct a LSTM network to learn the temporal features of sign language movements. The input data is time-series data, including keypoints extracted using MediaPipe Holistic: face, hands, and pose. The features of each frame are flattened into a 1,662-dimensional vector. The model receives 30 frames of sequence data, with an input shape of $(30, 1662)$. "1662" is the total number of dimensions for keypoints detected by MediaPipe Holistic per frame. Key points are expressed as follow:

- Face: 468 landmarks × 3D (x, y, z) = 1404
- Pose: 33 landmarks × 4 dimensions $(x, y, z, \text{visibility})$ = 132
- Left Hand: 21 landmarks × 3 dimensions (x, y, z) = 63
- Right Hand: 21 landmarks × 3 dimensions (x, y, z) = 63

These 1,662 numbers are flattened into a single vector and used as the input for one frame of the LSTM model. The model is built using TensorFlow/Keras' Sequential Application Programming Interface, and its architecture is based on Kamble et al.'s [4], consisting of a 64-unit LSTM in the first layer, which is set to pass the entire sequence to the subsequent layer and uses the activation function, a 128-unit LSTM in the second layer, which also passes the entire sequence and uses the activation function, and a 64-unit LSTM in the third layer, which outputs only the final feature vector and also uses activation function. These are stacked, and subsequently, a fully connected layer of 64 units and a 32-unit Dense layer, both with activation functions, are stacked. The final output layer is a Dense layer with a softmax activation function and the number of units set to 11 (the number of word classes) to calculate the probability of each word.

C. Resnet50 model for learning hand shapes

To identify groups of words with similar actions, we adopt ResNet50 [5] as a spatial feature model to determine the shape of the hand. First, we apply Arbitrary-Hands-3D-Reconstruction (ACR) [6], a 3D hand shape reconstruction model, to each video key frame to extract 3D parameters (.pkl files). Next, we process these parameters using the manopth library [7], [8] to reconstruct 3D mesh models (.obj files) of the left and right hands. Finally, we create a still image dataset for spatial feature training by rendering (photographing) this 3D hand model from multiple angles. This results in 185 3D model images of the right hand and 185 3D model images of the left hand, for a total of 370 hand model images, which we use to train ResNet50. Multi-class classification, which trains all 11 classes of shapes at once, risks averaging out slight differences in shape between classes. Therefore, we adopt a pairwise (one-to-one) learning strategy to specialize in identifying the shape differences between two specific words.

D. Testset evaluation

For the evaluation in this study, we prepare a test set of 30 new hand sign videos for each of the 11 words, separate from the training set, and use these to identify words. When a test video is input, the system first simultaneously obtains a time vector (keypoint sequence) for the LSTM and a 3D model for ResNet50 (for still image rendering).

In the first stage of inference, the temporal vector is fed into the LSTM model, and the classification probabilities (confidence scores) for the 11 classes are calculated using the Softmax function. The Softmax function is a function that converts the model's final layer outputs (logits), z, into a probability distribution, $\hat{y}$, totaling 1. Given K as the number of classes (11 in this paper), the probability $\hat{y}_k$ for class k is calculated by

$$\hat{y}_k = \frac{e^{z_k}}{\sum_{i=1}^{K} e^{z_i}}. \tag{1}$$

Next, a conditional correction using ResNet50 is executed based on the highest confidence score calculated by the LSTM.

- If Confidence $\geq$ 90%: The system determines that the LSTM's prediction has sufficient confidence and adopts its result as the final prediction.
- If Confidence $<$ 90%: The system determines that the LSTM is "hesitating" in its classification (e.g., the probability is dispersed among similar words). It then executes a correction using the spatial features from ResNet50.

In the correction process, the LSTM's probability vector P_{lstm} and the probability vector P_{resnet} (obtained from the pairwise ResNet50 classifiers) are integrated using Weighted Averaging to make a final decision. The final probability P_{final} is calculated using

$$P_{final} = w_{lstm} \cdot P_{lstm} + w_{resnet} \cdot P_{resnet}. \tag{2}$$

Here, w_{lstm} and w_{resnet} are coefficients representing the trust weight for each model's output, satisfying

$$w_{lstm} + w_{resnet} = 1. \tag{3}$$

We set $w_{lstm} = 0.3$ and $w_{resnet} = 0.7$.

The word classification accuracy is evaluated using Sensitivity (S_v). Sensitivity is a metric that indicates how correctly a specific class is predicted as positive (True Positive) and is calculated by

$$S_v = \frac{t_1}{t_1 + y_2}, \tag{4}$$

Where t_1 represents the count of True Positives and y_2 represents the count of False Negatives.

IV. Experimental Results

The performance of the two models is compared using the test set. The first model is an "LSTM-only" baseline model that uses only MediaPipe keypoint sequences (temporal features) as input. The second model is our proposed method, "LSTM + ResNet50," which uses hand shape information (spatial features) from ResNet50 for weighted average correction when the LSTM prediction confidence is below 90%. The sensitivity of each model is shown in TABLE I and TABLE II.

TABLE I
SENSITIVITY OF LSTM ONLY

Action	Correct	Total	Accuracy
again	24	30	80.00%
angry	27	30	90.00%
available	25	30	83.33%
bad	30	30	100.00%
bicycle	23	30	76.67%
big	28	30	93.33%
but	27	30	90.00%
car	28	30	93.33%
cheap	30	30	100.00%
clothes	30	30	100.00%
cold	30	30	100.00%
OVERALL	**302**	**330**	**91.52%**

As shown in TABLE I, the overall Sensitivity for the model using "LSTM only" is 91.52%. While it achieve 100% accuracy for four words including "bad" and "cheap", but is confirmed by recognition rates for words like "bicycle" (76.67%), "again" (80.00%), and "available" (83.33%).

TABLE II
SENSITIVITY OF LSTM + RESNET50

Action	Correct	Total	Accuracy
again	27	30	90.00%
angry	27	30	90.00%
available	28	30	93.33%
bad	30	30	100.00%
bicycle	26	30	86.67%
big	28	30	93.33%
but	30	30	100.00%
car	28	30	93.33%
cheap	30	30	100.00%
clothes	30	30	100.00%
cold	30	30	100.00%
OVERALL	**314**	**330**	**95.15%**

As shown in TABLE II, the overall Sensitivity improves to 95.15% with the proposed method "LSTM + ResNet50", which performs correction using spatial features via ResNet50. Specifically, the Sensitivity for "bicycle" improves from 76.67% to 86.67%, "again" improves from 80.00% to 90.00%, and "available" improves from 83.33% to 93.33%. Additionally, "but" also achieves 100% accuracy. As an example, we analyze the confusion pattern of the "again" class. The baseline LSTM model shows confusion, misclassifying "again" as either "bicycle" or "angry." Our ResNet50 correction method successfully distinguishes between "again" and "bicycle" by recognizing the difference in hand shape, correcting the final prediction to "again." However, misclassification between "again" and "angry" remains, even with the hybrid method, resulting in a final judgment of "angry".

V. Conclusion

This paper has addressed the misclassification problem encountered in recognizing sign language words with similar motion trajectories, a challenge highlighted in previous works [2]. We have proposed a hybrid architecture combining the LSTM network for temporal feature extraction with a ResNet50 model specifically trained using a pairwise strategy for spatial feature extraction. The core idea is that even if the movement is similar, the hand shape often holds crucial discriminating information. The baseline LSTM model achieves an overall sensitivity of 91.52% on the 11-word test set. By incorporating the ResNet50 hand shape classifier as a correction when the LSTM's confidence is below 90%, the overall sensitivity improves significantly to 95.15%. This is particularly true for words such as "bicycle," "again," and "available," which are difficult to recognize using LSTM alone.

Future work focuses on adjusting the weighted average that makes the final decision, improving the LSTM and Resnet50 systems, and validating the hybrid architecture on more extensive and challenging datasets, such as the 40 double-hand dynamic ASL words used by Abdullahi et al. [2] and the 72-word dataset including SRM words from Chophuk et al. [3], with the specific goal of comparing the accuracy of our proposed method against the accuracies reported in those studies, thereby confirming its generalization capabilities, particularly for SRM word groups and other known difficult cases.

References

[1] Abdullahi, S.B.; Chamnongthai, K. American Sign Language Words Recognition using Spatio-Temporal Prosodic and Angle Features: A sequential learning approach. IEEE Access 2022, 10, 15911–15923.

[2] S. B. Abdullahi and K. Chamnongthai, "American Sign Language Words Recognition of Skeletal Videos Using Processed Video Driven Multi-Stacked Deep LSTM," *Sensors*, vol. 22, no. 4, p. 1406, Feb. 2022.

[3] P. Chophuk, K. Chamnongthai, and K. Chinnasarn, "Backhand-approach-based American Sign Language words recognition using spatial-temporal body parts and hand relationship patterns," *Appl. Sci.*, vol. 11, no. 21, p. 9934, 2021.

[4] S. Kamble, "SLRNet: A Real-Time LSTM-Based Sign Language Recognition System," *arXiv preprint arXiv:2506.11154 [cs.CV]*, 2025.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[6] Z. Yu et al., "ACR: Attention collaboration-based regressor for arbitrary two-hand reconstruction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2021, pp. 11574–11583.

[7] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *ACM Trans. Graph.*, vol. 36, no. 6, Article 246, 2017.

[8] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "MANO: A model and library for hand-o-object interaction," *GitHub repository*, 2019. [Online]. Available: https://github.com/hassony2/manopth