ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

SIGN LANGUAGE RECOGNITION BASED ON GEOMETRIC FEATURES USING DEEP LEARNING

Eko Mulyanto Yuniarno

Computer Engineering Department, Institut Teknologi Sepuluh Nopember

email: ekomulyanto@ee.its.ac.id

Sign language plays a crucial role in facilitating communication among individuals with hearing impairments. In Indonesia, the deaf community often rely on BISINDO (Indonesian Sign Language) to communicate amongst themselves. People who are unfamiliar with sign language will face difficulties. This research aims to develop a system for recognizing sign language using geometric features extracted from hand joint coordinates using Google's MediaPipe Hands framework. The dataset contains 12 common words, each recorded 30 times with 30 frames recorded for each instance. This will facilitate communication between deaf and hearing individuals. We conducted tests on LSTM- Geometric and CNN1D- Geometric models using geometric features, and on CNN-LSTM-Spatial and CNN1D-LSTM-Spatial models using spatial features. The results indicate that the LSTM model with geometric features achieved the highest accuracy of 99%. Geometric features have been shown to be more effective than spatial features for classifying sign language gestures.

Keywords : Sign Language Recognition, Geometric Features, Spatial Feature

Received: 04-07-2024 | **Revised:** 15-07-2024 | **Accepted:** 22-07-2024
DOI: https://doi.org/10.23887/janapati.v13i2.82103

INTRODUCTION

Sign language plays a vital role in the lives of people with hearing impairments. In Indonesia, the deaf community commonly uses BISINDO (Bahasa Isyarat Indonesia) to communicate [1]. This communication involves precise hand and finger movements to convey words and phrases. Individuals with normal hearing without an understanding of sign language often encounter difficulties communicating with deaf individuals during their day-to-day interactions [2]. This communication barrier can lead to social isolation for those who are deaf or hard of hearing, significantly affecting their overall well-being [3]. Various sign language recognition technologies have been developed to address this challenge and facilitate more inclusive interactions between people with or without hearing and the broader community.

Numerous devices have been developed to translate sign language into spoken or written language, aiming to address these challenges [4][5][6]. A common approach to sign language recognition is computer-vision-based technology [7]. This technology eliminates the need for additional body sensors, offering users a non-intrusive experience. In contrast to gloves, computer vision technology avoids disrupting users with extra devices, enabling seamless interaction [8]. Using computer vision algorithms, these systems can accurately interpret gestures and movements, facilitating real-time translation of sign language into text or speech and fostering a more natural and intuitive communication process that bridges the gap between the signing community and the general population [9].

Computer vision-based sign language recognition involves using a camera to capture and extract gestures in real time, extracting features, and recognizing the gestures [9]. Some researchers have developed color-feature-based gesture recognition methods, such as RGB color space [10], which is then classified using machine learning models such as SVM [11][12][13], MLP, and deep learning [14][15][16].

To enhance classification accuracy, skin color segmentation was utilized to distinguish the hand from the background. The skin color extracted from the hand is separated using color spaces such as the Hue value of the HSV color model and YCbCr. This process aims to enhance the precision of sign language recognition [17][18][19].

Spatial data features, like pixel coordinates, are greatly affected by changes in orientation, translation, and scale [20]. Addressing this challenge requires a large dataset that covers a wide range of scale and orientation combinations to construct reliable machine-learning models. Advancements in pose estimation technologies, such as MediaPipe [21] and OpenPose [22], have led to the development of geometric features in the form of landmark coordinates from hand joints.

To mitigate the influence of spatial data, some researchers have developed features derived from inter-joint coordinates, such as angle, segment length, and normalized coordinate features [23][24]. However, these derived features lack spatial information, which can hinder accurate gesture recognition in changing orientations.

In this paper, we propose a new geometric approach that combines the hand's joint-angle features with the orientation features. This research proposes hand geometric features. These features are extracted from the coordinates of the finger joints. Twenty-one joint points are involved, resulting in 19 joint angles. These features exclude spatial information; thus they are invariant to orientation, translation, and scale changes.

METHOD

A. Geometric features

The proposed geometric features were extracted from the coordinates of the hand joints. We used Google's MediaPipe Hands framework to track hand positions using joint coordinate landmarks. The framework provides three outputs: coordinates of landmark positions, a detection score that indicates the model's confidence in hand detection, and Handedness Classification, which identifies whether it's the left or right hand.

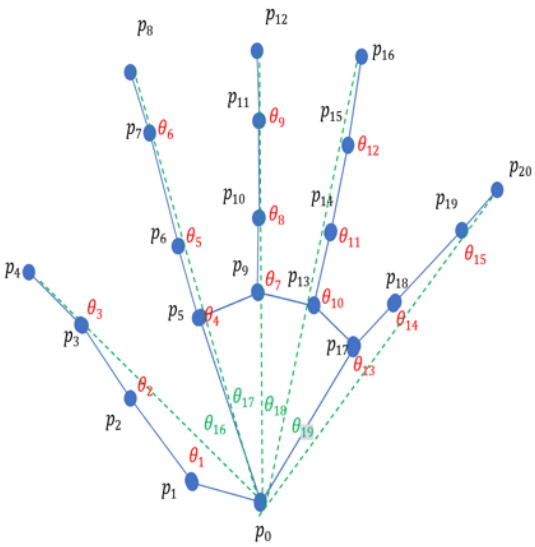


Figure1. Hand Landmark Position and Angles Joint

The geometric features developed for this study were derived from the angles between the hand and joints. The selected hand joint angles are shown in Figure 1, consisting of 19 angles, denoted as θ. θ_1 to θ_{15} are the angles at the finger joints and θ_{16} to θ_{19} The angles between the fingers to handle simultaneous opening and closing movements of the fingers.

The geometric feature extraction is as follows:

Let P be the set of landmark coordinates corresponding to Table 1.

$$P = (p_0, p_1, \cdots, p_{20}) \quad (1)$$

With $p_i \in P$,i $i \cdots 20$ are the landmark coordinates pixels. Therefore, the angles between the joints used as features are shown in Eq (2)

$$\begin{aligned}
\theta_1 &= \angle p_0 p_1 p_2 \\
\theta_2 &= \angle p_1 p_2 p_3 \\
\theta_3 &= \angle p_2 p_3 p_4 \\
\theta_4 &= \angle p_0 p_5 p_6 \\
\theta_5 &= \angle p_5 p_6 p_7 \\
\theta_6 &= \angle p_6 p_7 p_8 \\
\theta_7 &= \angle p_0 p_9 p_{10} \\
\theta_8 &= \angle p_9 p_{10} p_{11} \\
\theta_9 &= \angle p_{10} p_{11} p_{13} \\
\theta_{10} &= \angle p_0 p_{13} p_{14} \\
\theta_{11} &= \angle p_{13} p_{14} p_{15} \\
\theta_{12} &= \angle p_{14} p_{15} p_{16} \\
\theta_{13} &= \angle p_0 p_{17} p_{18} \\
\theta_{14} &= \angle p_{17} p_{18} p_{19} \\
\theta_{15} &= \angle p_{18} p_{19} p_{20} \\
\theta_{16} &= \angle p_4 p_0 p_8 \\
\theta_{17} &= \angle p_8 p_0 p_{12} \\
\theta_{18} &= \angle p_{12} p_0 p_{16} \\
\theta_{19} &= \angle p_{16} p_0 p_{20}
\end{aligned} \quad (2)$$

Then, global orientation features are added to address gestures involving hand orientation. The global orientation features represent the angles of vectors. $v_1 = \overline{p_0 p_5}$, $v_2 = \overline{p_0 p_{17}}$, and $v_3 = \overline{p_{17} p_5}$ relative to vector $v_x = (1,0)$, which is the direction vector of the x-axis. These three angles improve the gesture recognition accuracy.

$$\theta_{20} = \text{acos}\left(\frac{v_x \cdot v_1}{|v_1|}\right) \quad (3)$$

$$\theta_{21} = \text{acos}\left(\frac{v_x \cdot v_2}{|v_2|}\right) \quad (4)$$

$$\theta_{22} = \text{acos}\left(\frac{v_x \cdot v_3}{|v_3|}\right) \quad (5)$$

The joint angles in Eq. (2) and the global orientation angles in Eq. (3,4,5) are then combined to form the geometric features of $\mathbf{\Theta}$.

$$\mathbf{\Theta} = (\theta_1, \theta_2, \cdots, \theta_{22}) \quad (6)$$

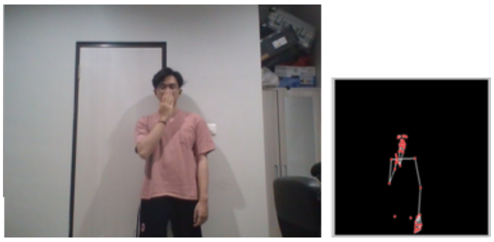
In geometric-based gesture recognition for sign language, an additional feature is implemented to detect the presence of hands in the frame as visibility features to indicate whether the hand is found.

$$\mathbf{V} = (v_L, v_R) \quad (7)$$

where $\mathbf{V}$ id visibility feature, v_L and v_R denote the visibility of the left and right hands, respectively, 1 indicates the presence of the hand, and 0 indicates its absence.

B. Spatial Features.

A spatial feature is constructed to facilitate comparison between geometric and spatial features. This image has joint coordinates connected by lines on a black background, as shown in Figure 2. b.



(a) (b)

Figure 2. Shows The Relation of The Original Image (a) and Its Spatial Feature (b).

In Figure 2, the original (Figure 2.a) is shown alongside the spatial features extracted from the image (Figure 2.b). The figure depicts a person visually performing a movement, with Figure 2.b showing the spatial features as joint coordinates connected by lines on a black background to indicate a particular pose pattern.

C. Data Set

The dataset comprises 12 fundamental words used in BISINDO sign language for basic communication. The words were divided into three categories[25].

1. **Words related to self and others**: This category is essential because it enables individuals to convey their identities and perspectives, thereby facilitating effective communication.
2. **Words related to interpersonal relationships**: This category is vital for forming emotional connections, expressing feelings, resolving conflicts, and showing empathy, all of which shape the emotional dynamics in relationships.
3. **Words related to possessions**: This category is crucial as it helps articulate and define property rights and boundaries, fostering clarity and preventing misunderstandings in interpersonal interactions.

Table 1. Words Grouped by Category

Category	Selected Word
Self and others	Saya (me), kamu (you) , siapa (who), nama(name)
interpersonal relationships	Tolong (help), terimakasih (thank you), Maaf (sory), dimana (where) , berapa (how many)
Possessions	Barang (stuff), ramah (house), ini (this)

The word categories are systematically presented in Table 1, providing a clear and organized overview.

The vocabulary used in the table has been selected based on an analysis of the variation in gestures. Where each word is associated with a different gesture.

For instance, the "thank you" gesture begins with the hand being placed over the mouth, with the palm facing the mouth, and then extended forward with the palm facing forward. In contrast, the "me" gesture begins with the hand being placed on the chest and slightly patted. The "this" gesture is characterized by the hand being positioned slightly above the stomach and the fingers pointing downwards.

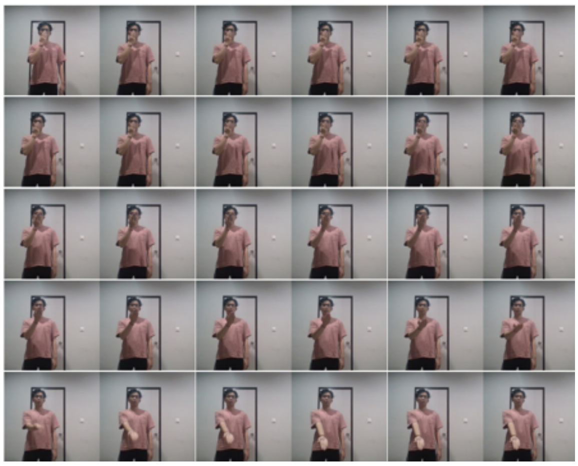


Figure 3. The Sequence of 30 Images Represent The "Terimakasih" (Thank You) Gesture.

Subsequently, one sign language expert and two individuals proficient in BISINDO sign language would proceed to demonstrate the appropriate sign for each selected word. Each word was recorded 30 times by the camera per person, with each recording consisting of 30 frames. Therefore, the total number of gestures to be classified was 1080, resulting in 32,400 images. Figure 3 shows an example of a series of 30 images representing the gesture for the word "terimakasih" (thank you).

D. Gesture Class

The gestures are divided into twelve classes in the following order: "barang", "terimakasih", "siapa", "rumah", "maaf", "ini", "tolong", "saya", "nama", "kamu", "dimana", "berapa". Based on this sequence, each class is obtained.

$$c_i = \left(\chi_i(1), \chi_i(2), \dots, \chi_i(n)\right) \quad (8)$$

where c_i is the class related to gesture ith, n is the number of classes, and χ_i is

$$\chi_i(j) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i \end{cases} \quad (9)$$

Thus, c_iIs a vector of length 12, and the index position with a value of 1 is the class of the gesture.

E. Sign Language Recognition based on Geometric Features

The block diagram in Figure 4 shows the steps required to perform geometric feature-based classification. Here, the dataset gesture is *F*, which is a set of image sequences representing a gesture comprising 30 frames.

$$F = (f_1, f_2, \cdots, f_N) \quad (10)$$

Where N is the number of frames.

Then, a set of right-hand (P_R) and left-hand (P_L) landmarks coordinates are extracted from F.

$$P_L = (p_{L,0}, p_{L,1}, \cdots, p_{L,20}) \quad (11)$$
$$P_R = (p_{R,0}, p_{R,1}, \cdots, p_{R,20}) \quad (12)$$

Then, the geometric features for the right and left hands are calculated from P_L and P_R based on Eq. (6). We include the visibility feature for each frame.

$$\mathbf{\Theta}_L = \left(\theta_{L,1}, \theta_{L,2}, \cdots, \theta_{L,22}\right) \quad (13)$$
$$\mathbf{\Theta}_R = \left(\theta_{R,1}, \theta_{R,2}, \cdots, \theta_{R,22}\right) \quad (14)$$
$$\mathbf{V} = (v_1, v_2) \quad (15)$$

Next, these features are classified using two deep learning models, namely LSTM-Geometric and CNN1D-Geometric, whose configurations are shown in Figures 5 and 6, respectively.

The input to both models comprised three types of input: $InputLeft$, $InputRight$, and $InputVisibility$. These three inputs are represented by Θ_L, Θ_R and V as follow

$$InputLeft = \left(\mathbf{\Theta}_{L,1}, \mathbf{\Theta}_{L,2}, \cdots, \mathbf{\Theta}_{L,N}\right)^T \quad (16)$$
$$InputRight = \left(\mathbf{\Theta}_{R,1}, \mathbf{\Theta}_{R,2}, \cdots, \mathbf{\Theta}_{R,N}\right)^T \quad (17)$$
$$InputVisibility = (\mathbf{V}_1, \mathbf{V}_2, \cdots, \mathbf{V}_N)^T \quad (18)$$

Where $\Theta_{L,i}$ and Θ_{Ri}Are the geometric features of the left and right hands of ith frame, $\boldsymbol{V_i}$ is the visibility of the related hand.

The LSTM-Geometric model processes each input sequentially by a time-distributed dense layer, followed by a dropout layer. The outputs are merged into a single tensor in a concatenate layer. LSTM captures the temporal dependencies in the data, and a final dense layer classifies the gestures into 12 classes .

The CNN1D-Geometric model processes the input via a Conv1D layer to extract features, followed by a MaxPooling1D layer to reduce the number of dimensions.

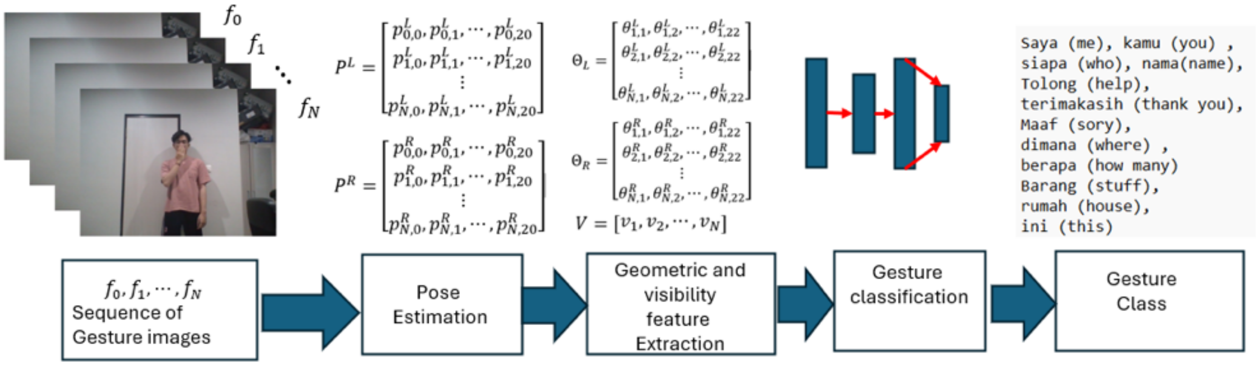


Figure 4. Block Diagram of Sign Language Recognition Based on Geometric Feature.

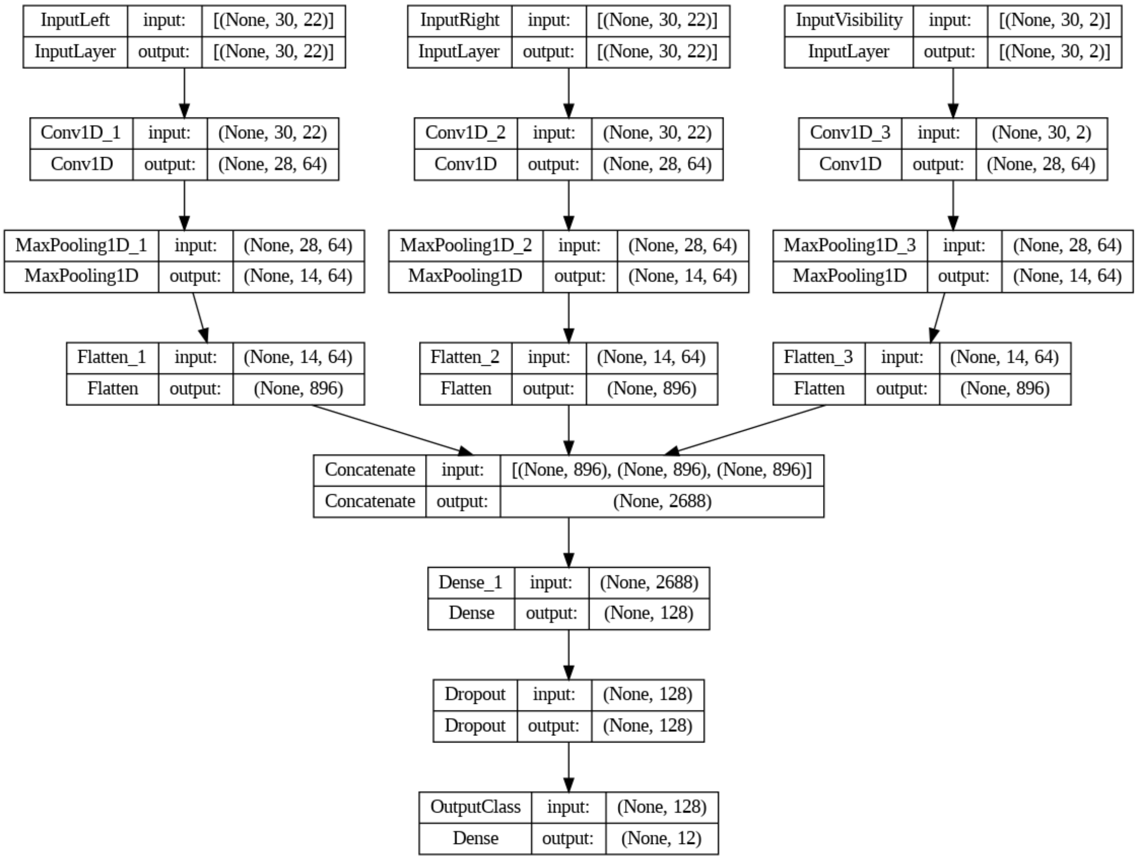


Figure 5. CNN1D-Geometric Model

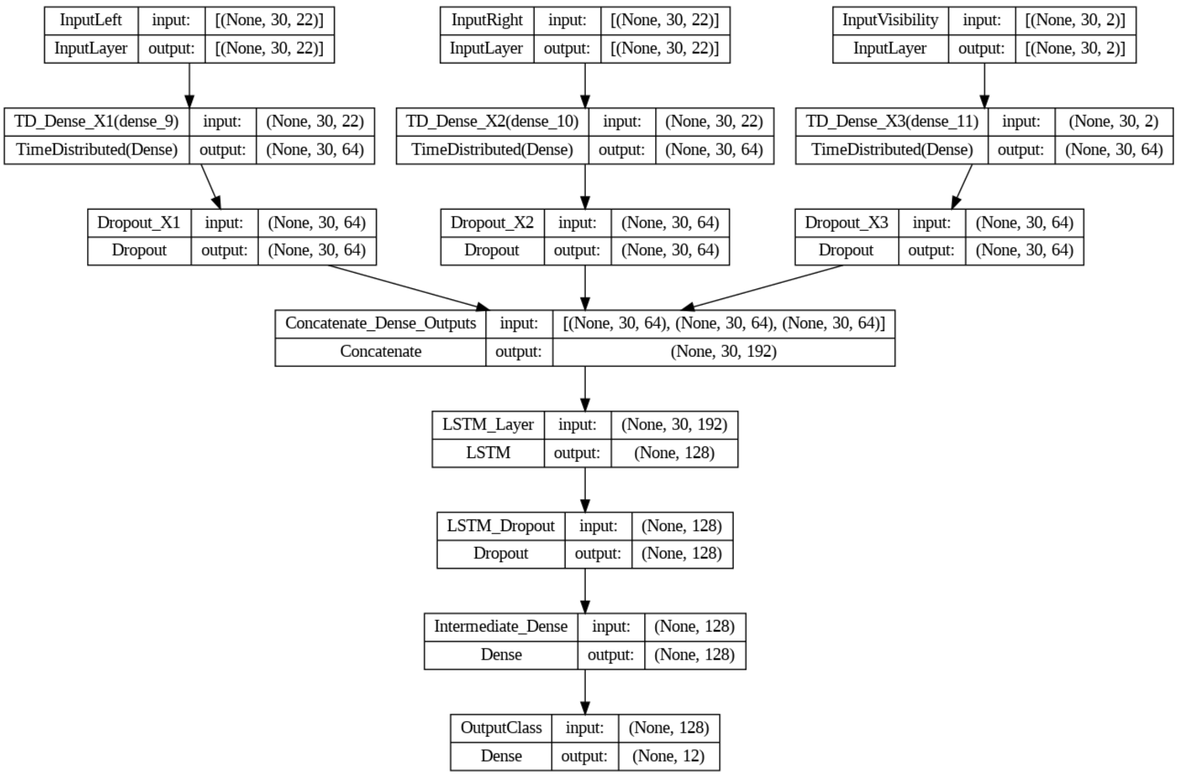


Figure 6. LSTM-Geometric Model

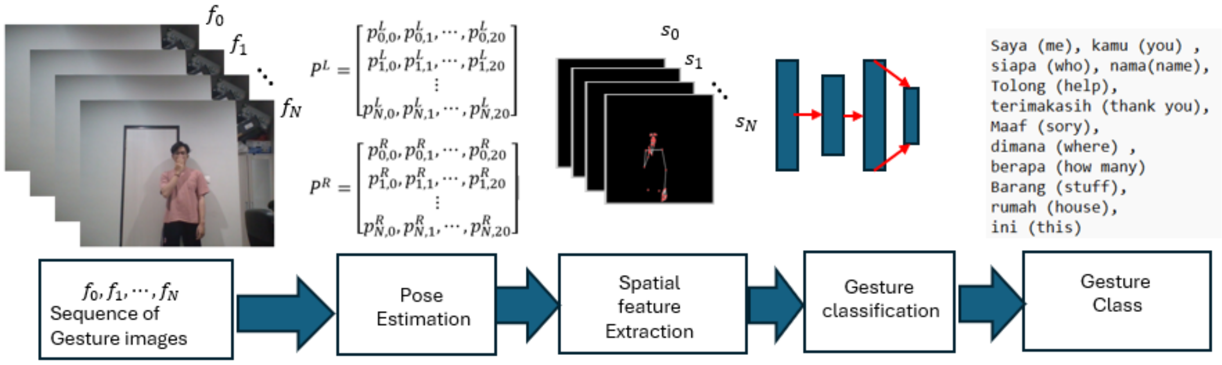


Figure 7. Block Diagram of Sign Language Recognition Based on Spatial Features

The output of MaxPooling1D was flattened into a vector. The vector is combined into a single tensor using a concatenate layer and then processed by a dense layer to classify the data using dropout to prevent overfitting. Finally, a dense layer produces 12 gestures classes.

The output is obtained by applying c_i in equation (8 to each of the two models LSTM-Geometric and CNN1D-Geometric.

F. Sign language recognition based on spatial features.

Figure 7 shows a block diagram of gesture classification based on spatial features for sign language. The process comprises five steps, with the initial two steps corresponding to gesture frame extraction and pose estimation, which are analogous to the initial two steps in spatial geometric feature-based classification. The distinguishing factor is that the selected feature is the pose image. s_i with $i = 0, .., N$, representing the pose image in frame f_i

The spatial-feature-based gesture classification begins by capturing a sequence of gesture images labeled as $F = f_1, f_2, \ldots, f_N$. For each image f_i , the pose is estimated to acquire the coordinates of body key points denoted as $P_i = \{p_{i1}, p_{i2}, \cdots, p_{iM}\}$ With, M represents the number of key points identified. If there exists a graph G is an ordered pair, then we have

$$G_i = (P_i, E) \quad (19)$$

This represents the relationship of each point in P_i with $E \subseteq p_i, p_j$ | $p_i, p_j \in P$ dan $p_i \neq p_j$ is the set of edges.

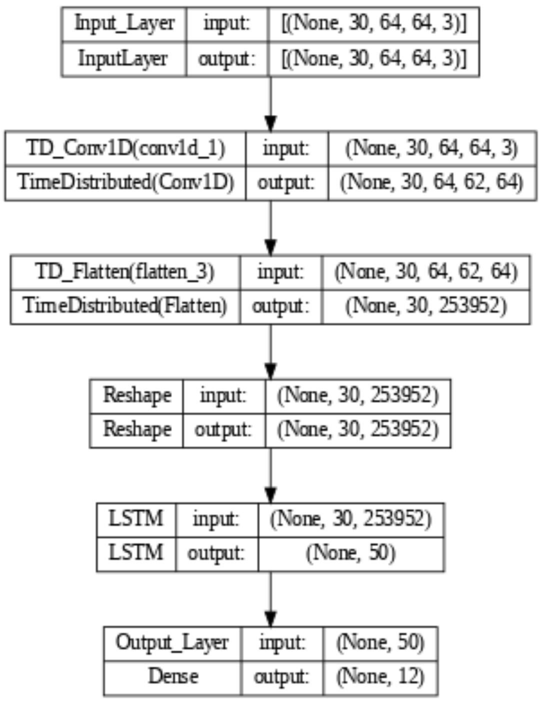


Figure 8. CNN1D-LSTM-Spatial Model

Based on eq.19 We obtain a spatial feature.

$$s_i = G_i \quad (20)$$

Let $S = s_0, s_1, \cdots, s_N$ is a set of spatial features. After the spatial feature is created, it is classified using the CNN-LSTM-Spatial and CNN1D-LSTM-Spatial models. whose configurations are shown in Figure 8 and Figure 9, respectively.

The CNN1D-LSTM-Spatial model combines a Convolutional Neural Network (CNN) and long short-term memory (LSTM) to process data of both spatial and temporal dimensions. The model begins with an image sequence of 64 × 64 pixels

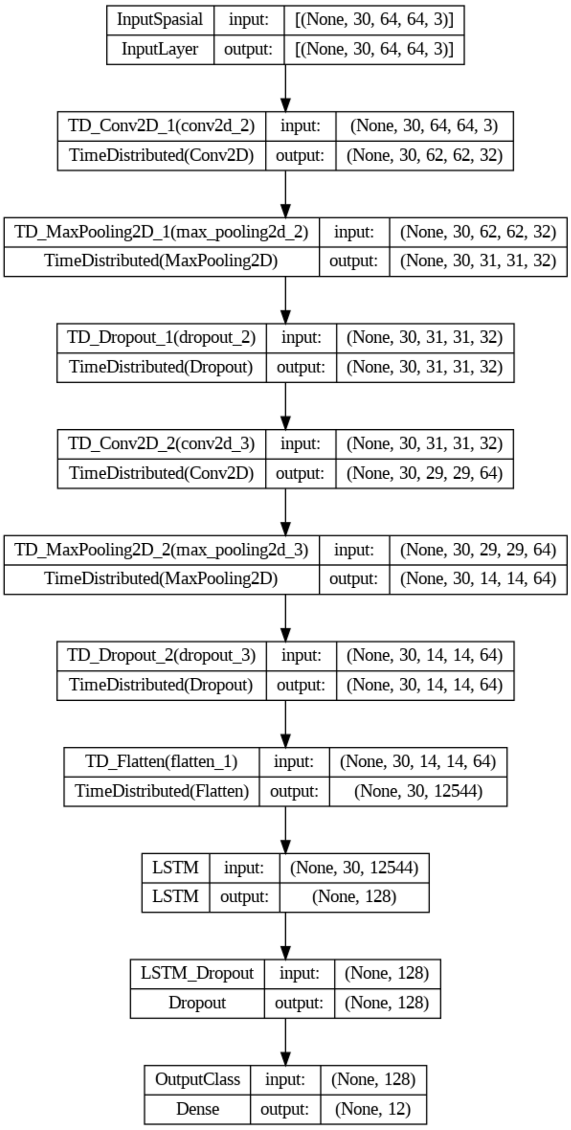


Figure 9. CNN-LSTM-Spatial Model

and three-color channels. The TimeDistributed Conv1D layer extracts spatial features from each image frame in the sequence. These results are flattened and reshaped for processing by the LSTM layer, which captures the temporal dependencies in the sequence.

The CNN-LSTM-Spatial model combines a Convolutional Neural Network (CNN) and long short-term memory (LSTM) to process data in space and time. The input layer receives an image sequence of dimensions (30, 64, 64, 3), and extracts spatial features from each image frame. This is achieved using a TimeDistributed Conv2D layer. Next, the spatial dimensions are reduced using a MaxPooling2D layer, and overfitting is prevented using a Dropout layer. This process is repeated with a second layer of convolutions and pooling, which results in more dense and reduced features.

CNN and LSTM process spatial and temporal sequence data. The input layer receives an image sequence with dimensions (30, 64, 64, 3) and extracts spatial features from each image.

Both models produce the same output, which is c_i as per equation (8), like the LSTM-Geometric and CNN1D-Geometric models.

G. Performance And Evaluation

The following metrics are used to evaluate the performance of a classification model[26]:

1. Accuracy: percentage of correct predictions out of total predictions.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{Total Predictions}} \tag{21}$$

2. Precision: percentage of correct positive predictions out of all positive predictions made.

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

3. Recall (Sensitivity or TPR-True Positive Rate), Definition: The percentage of correct positive predictions out of all true positive cases.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{23}$$

4. F1 Score: Harmonized average of precision and recall. It provides a balance between the two metrics.:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{24}$$

Here, TP = true positive, FP = false positive,TN = true negative, and FN = false negative.

RESULTS AND DISCUSSION

One issue identified was that the 1080 gestures were unable to detect all hands within each frame accurately. Table 2 illustrates the distribution of the number of frame that hands can be successfully identified within each gesture.

Out of 1080 gestures analyzed, the right hand was detected in less than 15 frames in 94 gestures, and the left hand in 79 gestures. Conversely, the right hand was detected in more than 15 frames in 986 gestures, and the left hand in 1001 gestures.

Table 2. Distribution of Frames with Detected Hands Per Gesture

Number of Frames	Right Hand	Left Hand
0 – 5	33	55
6-10	17	6
11-15	44	18
16-20	69	29
21-25	176	65
26-30	741	907
Total	1080	1080

The data indicate that a significant number of gestures are performed with incomplete hand landmark information. To solve this problem, the angle value corresponding to the landmark coordinates and the hand visibility value is then set to zero when the hand is not detected.

After inputting the missing data, the 1080 data set was split into two separate sets: 70% was used for training and the remaining 30% was set aside for validation. Subsequently, the validation data was used to evaluate how effective the features were in various models.

Figures 9 and 10 show the confusion matrix for gesture recognition using CNN-1D-geometric and LSTM-geometric. Both matrices show a strong diagonal with values from 25 to 27, supported by 27 instances. A geometric-based model leads to more precise classification, achieving 92% to 100% accuracy rates. However , Figures 11 and 12 show that the spatial feature-based model has a dominant diagonal confusion matrix with values between 13 and 26. It indicates that the accuracy is between 48% (13 out of 27) and 96% (26 out of 27).

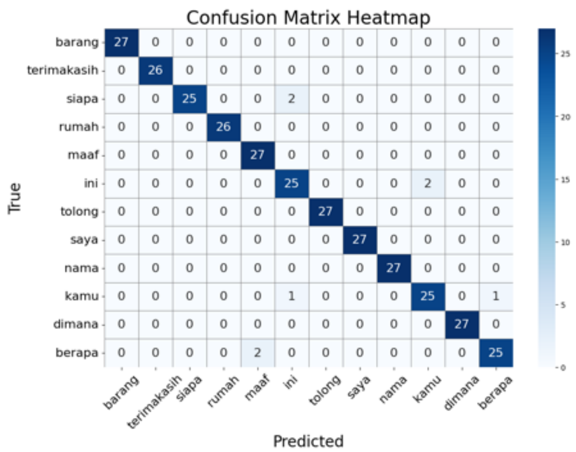


Figure 9. Confusion Matrix of CNN1D-Geometric Model

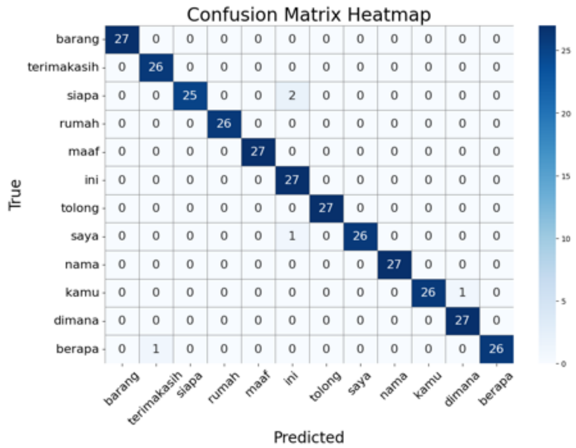


Figure 10. Confusion Matrix of LSTM-Geometric Model

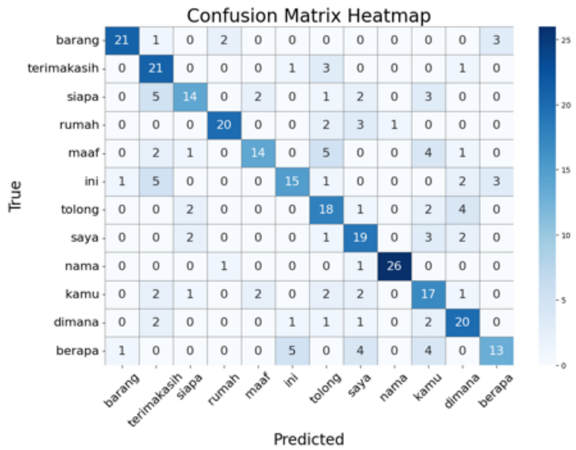


Figure 11 Confusion Matrix of CNN1D-LSTM - Spatial Model

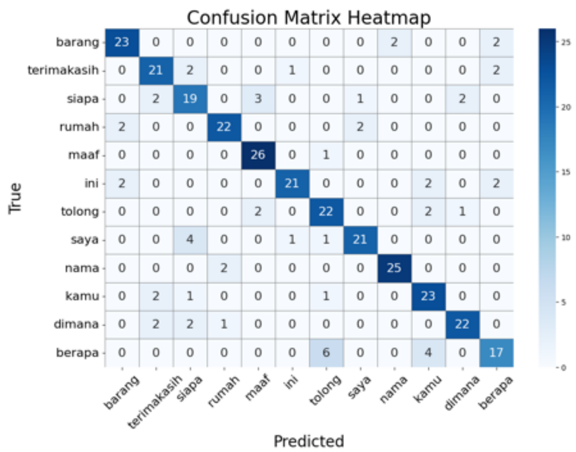


Figure 12. Confusion Matrix of CNN-LSTM Model

The overview of the confusion matrix is shown in Table 2. The accuracy, average precision, average recall, and average F1 score are shown from equations 21, 22, 23, and 24, representing the performance assessment of each model.

Table 3. Model Performance Evaluation

Model	Acc	Average Precision	Average Recall	Average F1-score
CNN1D-Geometric	0.98	0.98	0.98	0.98
LSTM-Geometric	0.98	0.99	0.98	0.98
CNN1D-Spatial	0.68	0.7	0.68	0.68
CNN-LSTM Spatial	0.81	0.82	0.81	0.81

The data presented in Table 3 demonstrates that the LSTM-Geometric model attained the highest accuracy percentage of 98%, suggesting that it accurately predicted outcomes for 98% of the total tests conducted. The CNN1D-Geometric model achieved an accuracy of 98%, while the CNN1D-Spatial model had a 68% accuracy, and the CNN1D-LSTM-Spatial model achieved an accuracy of 81%.

The LSTM-Geometric model's high precision of 0.98 shows its accurate identification of positive hand gestures, with rare misclassifications. The CNN1D-Geometric model also showed strong performance, achieving a precision of 0.98, demonstrating the efficiency of utilizing geometric features in recognizing hand gestures with minimal errors.

On the other hand, the model that used spatial features showed relatively lower effectiveness. The CNN1D-Spatial model showed a misclassification tendency with an average precision of just 0.70. On the other hand, the CNN1D-LSTM-Spatial model obtained an average precision of 0.82 despite being less effective than the model that utilized geometric features.

The LSTM geometric model and the CNN1D geometric model both have a recall rate of 98%. This indicates that both models are able to accurately recognize 98% of the positive signals. The CNN1D-Spatial model achieved a recall rate of 0.68, the lowest among all models. This model recognized just 68% of the favorable signals. The CNN1D-LSTM-Spatial model achieved an 81% recall rate, surpassing the CNN1D-Spatial model but falling short of the Geometric model.

The CNN1D-Geometric model performed well, with an F1 score of 97%. The CNN1D-Spatial and CNN1D-LSTM-spatial models obtained F1 scores of 64% and 87%, respectively. The results demonstrate that a model using geometric features is more accurate and reliable than a model using spatial features in recognizing hand signs in sign language.

CONCLUSION

This research highlights the advantages of geometric features over spatial features in sign language recognition. Quantitative results show that classification with geometric features performs better than classification based on spatial features.

The discussion shows that the performance of geometric feature-based models such as LSTM-Geometric achieves 98% accuracy, average precision of 0.99, average recall of 0.98, and average F1 score = 98. Another geometric feature-based model, CNN1D-Geometric, approaches this performance with 98 % accuracy, an average precision of 0.98, an average recall of 0.98, and an average F1 score of 0.98.

However, spatial feature-based classification shows that the CNN1D-spatial model has 68% accuracy, average precision of 0.70, average recall of 0.68, and average F1 Score of 0.68. In comparison, the CNN-LSTM-spatial model has better performance than CNN1D-spatial with an accuracy of 81%, average precision of 0.82, average recall of 0.81, and average F1 Score of 0.81.

The performance comparison shows that geometric features significantly improve the accuracy and consistency of classification in sign language recognition. These results demonstrate that, compared to spatial features, geometric features significantly enhance classification accuracy and consistency in sign language recognition.

REFERENCES

[1] Sihananto, A. N., Safitri, E. M., Maulana, Y., Fakhruddin, F., & Yudistira, M. E. (2023). Indonesian Sign Language Image Detection Using Convolutional Neural Network (CNN) Method. *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, 13(1), 13-21

[2] Bai, Y., & Bruno, D. (2020). Addressing Communication Barriers Among Deaf Populations Who Use American Sign Language in Hearing-Centric Social Work Settings. *Polymer Journal, 18*, 37-50.

[3] Luft, Pamela. (2000). Communication barriers for deaf employees: Needs assessment and problem-solving strategies. *Work (Reading, Mass.)*. 14. 51-59.

[4] Z. R. Saeed, Z. B. Zainol, B. B. Zaidan and A. H. Alamoodi, "A Systematic Review on Systems-Based Sensory Gloves for Sign Language Pattern Recognition: An Update From 2017 to 2022," in IEEE Access, vol. 10, pp. 123358-123377, 2022

[5] Y. -C. Lai, P. -Y. Huang and T. -S. Horng, "Wi-Fi SIMO Radar for Deep Learning-Based Sign Language Recognition," in IEEE Microwave and Wireless Technology Letters, vol. 34, no. 6, pp. 825-828, June 2024
[6] Chong T-W, Lee B-G. American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach. Sensors. 2018; 18(10):3554.
[7] T. Li, Y. Yan and W. Du, "Sign Language Recognition Based on Computer Vision," 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2022
[8] Westlund, J. K., D'Mello, S. K., & Olney, A. M. (2015). Motion Tracker: Camera-Based Monitoring of Bodily Movements Using Motion Silhouettes. *PLOS ONE, 10*(6), e0130293
[9] Aloysius, N., & Geetha, M. (2020). Understanding vision-based continuous sign language recognition. *Multimedia Tools and Applications, 79*(33), 22177-22209
[10] D. Fan, H. Lu, S. Xu and S. Cao, "Multi-Task and Multi-Modal Learning for RGB Dynamic Gesture Recognition," in IEEE Sensors Journal, vol. 21, no. 23, pp. 27026-27036, 1 Dec.1, 2021
[11] M. A. A. Razak, F. Y. A. Rahman, R. Mohamad, S. Shahbuddin, Y. W. M. Yusof and S. I. Suliman, "Hand Gesture Recognition based on Convolution Neural Network (CNN) and Support Vector Machine (SVM)," 2023 IEEE 14th Control and System Graduate Research Colloquium (ICSGRC), Shah Alam, Malaysia, 2023, pp. 123-126.
[12] X. -R. Song, J. Yang, S. -M. Zhang and S. Gao, "Research on Gesture Recognition Method Based on Image Processing and SIFT-SVM," 2021 China Automation Congress (CAC), Beijing, China, 2021, pp. 526-531.
[13] P. Pruthvi and J. Geetha, "Convolution Neural Network for Predicting Alphabet Sign Language and Comparative Performance Analysis of CNN, KNN, and SVM Algorithms," 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2022, pp. 1-6.
[14] T. Shanableh, "Two-Stage Deep Learning Solution for Continuous Arabic Sign Language Recognition Using Word Count Prediction and Motion Images," in IEEE Access, vol. 11, pp. 126823-126833, 2023, doi: 10.1109/ACCESS.2023.3332250.
[15] J. Shin, A. S. M. Miah, K. Suzuki, K. Hirooka and M. A. M. Hasan, "Dynamic Korean Sign Language Recognition Using Pose Estimation Based and Attention-Based Neural Network," in IEEE Access, vol. 11, pp. 143501-143513, 2023.
[16] O. Koller, N. C. Camgoz, H. Ney and R. Bowden, "Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Discover Sequential Parallelism in Sign Language Videos," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 9, pp. 2306-2320, 1 Sept. 2020.
[17] A. Agarwal, R. Sreemathy, M. Turuk, J. Jagdale and V. Kumar, "Indian Sign Language Recognition using Skin Segmentation and Vision Transformer," 2023 IEEE 20th India Council International Conference (INDICON), Hyderabad, India, 2023, pp. 857-862, doi: 10.1109/INDICON59947.2023.10440818.
[18] S. Reshna and M. Jayaraju, "Spotting and recognition of hand gesture for Indian sign language recognition system with skin segmentation and SVM," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2017, pp. 386-390, doi: 10.1109/WiSPNET.2017.8299784.
[19] U. Rastogi, A. Pandey and V. Kumar, "Skin Segmentation and SVM for Identification and Spotlighting of Hand Gesture for ISLR System," 2023 6th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2023, pp. 1-5.
[20] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision, 60*(2), 91-110.
[21] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M., Lee, J., Chang, W.-T., Hua, W., Georg, M., & Grundmann, M. (2019). MediaPipe: A Framework for Perceiving and Processing Reality. In *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*.
[22] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2019). OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1), 172-186.
[23] J. Shin, A. S. M. Miah, K. Suzuki, K. Hirooka and M. A. M. Hasan, "Dynamic Korean Sign Language Recognition Using Pose Estimation Based and Attention-Based Neural Network," in IEEE Access, vol. 11, pp. 143501-143513, 2023, doi: 10.1109/ACCESS.2023.3343404.

[24] Ko, S.-K., Kim, C. J., Jung, H., & Cho, C. (2019). Neural Sign Language Translation Based on Human Keypoint Estimation. *Applied Sciences*, 9(13), 2683

[25] Lathren, C.R., Rao, S.S., Park, J. et al. Self-Compassion and Current Close Interpersonal Relationships: a Scoping Literature Review. Mindfulness 12, 1078–1093 (2021). https://doi.org/10.1007/s12671-020-01566-5

[26] Naidu, G., Zuva, T., Sibanda, E.M. (2023). A Review of Evaluation Metrics in Machine Learning Algorithms. In: Silhavy, R., Silhavy, P. (eds) Artificial Intelligence Application in Networks and Systems. CSOC 2023. Lecture Notes in Networks and Systems, vol 724. Springer, Cham.