Journal of
Information Systems Engineering
and Business Intelligence

Vol.11, No.1, February 2025
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

Dynamic Sign Language Recognition in Bahasa using MediaPipe, Long Short-Term Memory, and Convolutional Neural Network

Ivana Valentina Lemmuela [1)], Mewati Ayub [2)*], Oscar Karnalim [3)]

[1)2)3)] *Faculty of Smart Technology and Engineering, Maranatha Christian University, Bandung, Indonesia*

[1)]ivanavlemmuela@gmail.com, [2)]mewati.ayub@it.maranatha.edu, [3)]oscar.karnalim@it.maranatha.edu

Abstract

Background: Communication is important for everyone, including individuals with hearing and speech impairments. For this demographic, sign language is widely used as the primary medium of communication with others who share similar conditions or with hearing individuals who understand sign language. However, communication difficulties arise when individuals with these impairments attempt to interact with those who do not understand sign language.

Objective: This research aims to develop models capable of recognizing sign language movements in Bahasa and converting the detected gesture into corresponding words, with a focus on vocabularies related to religious activities. Specifically, the research examined dynamic sign language in Bahasa, which comprised gestures requiring motion for proper demonstration.

Methods: In accordance with the research objective, sign language recognition model was developed using MediaPipe-assisted extraction process. Recognition of dynamic sign language in Bahasa was achieved through the application of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) methods.

Results: Sign language recognition model developed using bidirectional LSTM showed the best result with a testing accuracy of 100%. However, the best result for the CNN alone was 86.67 %. The integration of CNN and LSTM was observed to improve performance than CNN alone, with the best CNN-LSTM model achieving an accuracy of 95.24%.

Conclusion: The bidirectional LSTM model outperformed the unidirectional LSTM by capturing richer temporal information, with a specific consideration of both past and future time steps. Based on the observations made, CNN alone could not match the effectiveness of the Bidirectional LSTM, but a combination of CNN with LSTM produced better results. It is also important to state that normalized landmark data was found to significantly improve accuracy. Accuracy within this context was also influenced by shot type variability and specific landmark coordinates. Furthermore, the dataset containing straight-shot videos with x and y coordinates provided more accurate results, dissimilar to those comprised of videos with shot variation, which typically require x, y, and z coordinates for optimal accuracy.

Keywords: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), MediaPipe, Sign Language

Article history: Received 12 July 2024, first decision 17 January 2025, accepted 14 February 2025, available online 28 October 20XX

I. Introduction

Communication is a significant challenge for individuals with hearing and speech impairments. For this demographic, sign language serves as the primary and very important medium of communication. However, regardless of the fact that sign language has been observed to significantly facilitate interaction within the demographic, it remains largely unfamiliar to individuals without hearing impairments, including those who can hear and speak. Typically, sign language combines motions, hand gestures, and facial expressions to convey meaning [1][2]. This complexity creates communication barriers between individuals with hearing loss and those who can hear and speak. Considering this issue, the present research aims to develop models capable of recognizing sign language gestures in Bahasa to assist individuals with hearing impairments in communicating with hearing and speaking individuals.

Sign language can be categorized into two types namely static and dynamic. Static sign language consists of hand gestures without movement, while dynamic sign language includes gestures requiring motion. Some words can be accurately represented using static gestures, while others necessitate dynamic movements to convey respective meanings effectively [3].

Sign language recognition research has been conducted in various languages using a wide range of methodologies, with some even adopting distance metrics for classification. For instance, Al Rivan et al. [4] investigated recognizing

[*] Corresponding author

doi: http://dx.doi.org/10.20473/jisebi.11.1.17-29

the American Sign Language (ASL) alphabet, using the Histogram of Oriented Gradients (HOG) for feature extraction. In the exploration, classification was performed using the K-Nearest Neighbor (K-NN) algorithm, specifically applying Euclidean Distance, Manhattan Distance, and Chebyshev Distance. Other research works have explored the use of neural networks for sign language identification. For example, Al Rivan, Noviardy, and Trinanda [5] extended the research on ASL alphabet recognition by adopting HOG for feature extraction and introducing Artificial Neural Networks (ANN) for classification. ANN consists of three layers namely the input, hidden, and output layers. Convolutional Neural Networks (CNN) have also been used in sign language recognition. This is evidenced by the exploration of Pratama et al. [6]where an ASL recognition model was developed using CNN with Model E architecture as the primary filter, after which its performance was compared with AlexNet. The most recent publication referenced in this research adopted MediaPipe as a tool for extracting sign language features while continuing to use neural networks for recognition. Similar to the referenced publication, Abdulhameid et al. [7]conducted research on dynamic ASL sign language recognition, where the preprocessing method adopted included the use of MediaPipe to detect key body points and capture sign language movements. These key points, or landmarks, were stored in a Comma-Separated Values (CSV) file, after which the learning process was carried out using Long Short-Term Memory (LSTM) networks implemented through the TensorFlow and Keras libraries.

As previously stated, numerous investigations have explored sign language recognition across various languages. Several of these research efforts have focused specifically on recognizing sign language in Bahasa, primarily targeting static signs for the alphabet and numbers. For instance, Supria, Murti, and Khotimah [8] examined 26 alphabet signs in Bahasa, including both static and dynamic gestures for the letters J and Z. The research included the use of a Leap Motion Controller (LMC) to capture and visualize hand shapes in 3D animation, aiding in the acquisition of hand coordinate data. Regardless of the significance of the exploration, some signs proved challenging to recognize due to similarities in hand shapes. Gafar and Sari [9]similarly conducted research on recognizing Indonesian alphabet sign language using the Fuzzy KNN method. Mahfudi et al. [10]also adopted the Scale-Invariant Feature Transform (SIFT) method to recognize Bahasa sign language for numbers from 1 to 10. Five invariances were identified in the research using SIFT, which led to the achievement of an average matching accuracy based on a threshold of 0.65. Nurhayati, Eridani, and Tsalavin [11] expanded the scope of research by analyzing 29 classes, including 26 Indonesian sign language letters and three additional classifications including space, erase, and not classified, using CNN. Accordingly, Sholawati, Auliasari, and Ariwibisono [12] conducted a research on Indonesian alphabet sign language recognition, excluding the letters J and Z, using CNN. Al Rivan, Hartoyo, and Suryanto [13] also applied CNN to classify Indonesian alphabet sign language, using the Visual Geometry Group (VGG)-16 architecture, recognized for its multiple layers with kernel filters and incorporated AlexNet for transfer learning. CNN generally computes convolutions between the input and a filter, with feature extraction occurring in an unsupervised manner, meaning no predefined rules dictate the process. Thira et al. [14]further advanced research in this field by investigating alphabet sign recognition using CNN models with MobileNetV2, MobileNetV3Small, and MobileNetV3Large architectures.

Many studies on Indonesian sign language recognition focus primarily on static signs for the alphabet and numbers. There are relatively few studies that address dynamic sign language recognition. Anggita, Khotimah, and Suciati [15] conducted a study to recognize 20 words in Indonesian dynamic sign language. They captured data using Kinect 2.0 and applied the Dynamic Time Warping (DTW) method for recognition. While this method was successful in recognizing dynamic sign language, it is challenging to use due to the requirement of Kinect hardware. Al Fajri, Jayanta, and Wahyono [16] conducted a study on recognizing Indonesian dynamic sign language using Colored Motion History Images (Colored MHI) for data preprocessing and Convolutional Neural Networks (CNN) for recognition. They focused on 5 Indonesian dynamic sign language words with minimal movement.

Despite advancements in sign language recognition, particularly for static signs in the Indonesian alphabet and numbers, a significant gap remains in recognition of dynamic sign language in Bahasa. The majority of existing explorations require specialized hardware such as Kinect, which is expensive and limits scalability as well as widespread application. Additionally, some research focused only on minimal movement gestures, which are rare in real-world communication, where sign language often includes expressive and dynamic gestures. These constraints have significantly limited the practical applicability of current research. It is also important to elucidate that few research have explored the combined use of LSTM and CNN for recognizing dynamic and symbolic sign language in Bahasa.

The current research focuses on recognizing sign language from videos using LSTM and CNN. To achieve the stated objectives, three experiments were conducted, the first used only LSTM, the second adopted only CNN, and the third included a combination of both methods. Sign language videos used were manually collected, with each word recorded individually. After collection, the videos were then preprocessed, where images were extracted frame by frame. MediaPipe was then used to obtain landmark data from the performers' bodies, capturing the movement of sign language. The dataset was divided into training and testing sets, which were used to train and evaluate the developed model respectively. Subsequently, multiple experiments were conducted to fine-tune the models for optimal performance.

Building upon previous explorations, this research aims to recognize both dynamic and symbolic Bahasa sign language using only simple devices, ensuring practical implementation. Symbolic sign language movements serve as abbreviated gestures to represent specific concepts rather than spelling out words character by character. For example, the word pray is expressed by holding both hands together in a backward motion. By using LSTM and CNN, this research evaluates these methods both individually and in combination to determine the most effective method for sign language recognition. These methods have been identified as effective in capturing dynamic and complex nature of sign language gestures [7][16]. Additionally, MediaPipe is capable of significantly enhancing recognition accuracy by providing precise landmark detection, facilitating more accurate gesture interpretation. Once the most effective model is identified, it can be implemented in an application to recognize sign language and translate it into text for subtitles.

II. METHODS

Developing sign language recognition system requires several essential steps. At a high level, this research is structured into four key stages, as shown in Fig. 1. For the case research, the most frequently spoken words in videos of religious services on YouTube were selected as the primary dataset. It is important to state that while the dataset is based on Christian religious services, sign language gestures identified may be applicable to other religious contexts, particularly those with similar perspectives. For instance, certain gestures, such as prayer, possess commonalities across different religious traditions.

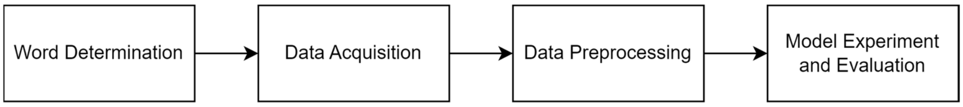


Fig. 1 Research Methodology

A. Word Determination

The first step in this research includes determining relevant words by identifying the most frequently spoken words in the observed Sunday service videos of religious activities on YouTube. The selection criteria required the videos to be recordings of religious services that included sign language interpretation to ensure accurate translation. Additionally, the videos had to provide accompanying scripts to facilitate precise word determination. Based on these criteria, a total of ten worship videos from the Gereja Mawar Sharon (GMS) Church YouTube channel, which features special sign language worship services, were selected.

The scripts from these ten videos were used as the dataset to extract the most frequently spoken words. The preprocessing phase comprised tokenizing the scripts into individual words. To refine the dataset, stop words in Bahasa were removed using a predefined stop-word library. Additionally, bigrams and trigrams were generated to capture two-word and three-word combinations, ensuring more meaningful phrase identification. The process included the combination of individual words into bigrams and trigrams, integrating each combination into a single dataset that included unigrams, bigrams, and trigrams. Subsequently, the 30 most frequently occurring words from the Sunday service videos were identified based on frequency. From this list, 15 words were manually selected based on domain knowledge. The selection was guided by the relevance of the words to religious activities and references in the Bible. The word determination process is shown in Fig. 2

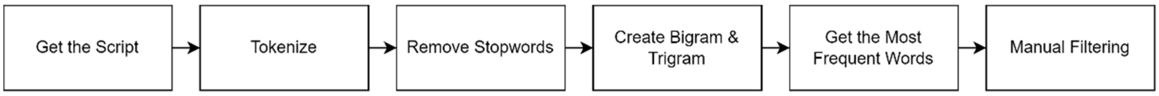


Fig. 2 Process Flow of Word Determination

B. Data Acquisition

For data acquisition, videos were recorded manually, each capturing a single sign language word. Three performers participated in the recordings, resulting in a total of 315 videos covering 15 words. It is important to comprehend that the dataset comprised a diverse range of shot types, including eye-level shots, close-up shots, low-angle shots, and wide shots. Specifically, 105 videos (equivalent to seven sets) featured all four shot variations (eye-level, close-up, low-angle, and wide shots), and the remaining 210 videos (equivalent to 14 sets) contained only eye-level and close-up shots. All videos were recorded in landscape orientation to maintain consistency in framing and perspective, after which each file was named according to the performed sign language word, followed by an underscore and a running number to distinguish individual recordings.

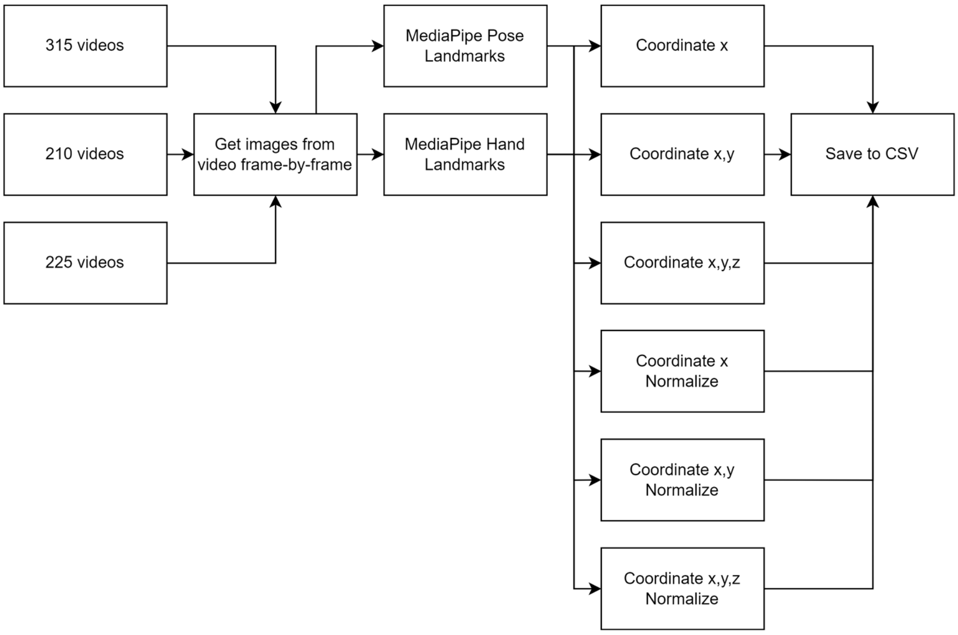


Fig. 3 Flow of Data Preprocessing

C. *Data Preprocessing*

The third step in the process is data preprocessing. In this phase, each video was extracted into individual frames, capturing images frame by frame. MediaPipe was then used to identify key points on the body, referred to as landmarks. This framework, which was developed by Google, was selected because it enables the application of Artificial Intelligence (AI) and Machine Learning (ML) across various platforms, including Android, Web, Python, and iOS. One of its capabilities is skeleton detection, which is essential for sign language recognition[17]. Several MediaPipe libraries were used in this research to enhance the accuracy of landmark extraction. The preprocessing workflow is presented in Fig. 3.

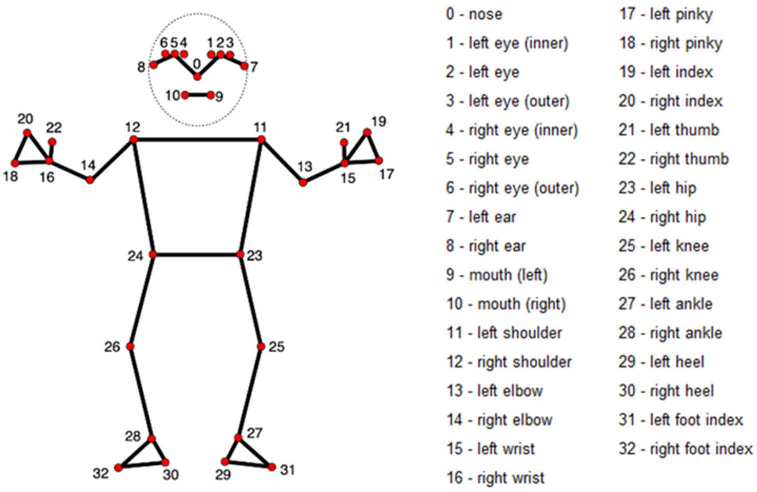


Fig. 4 Landmark Data from MediaPipe Pose Landmarks [18]

First, MediaPipe Pose Landmark was used to detect body landmarks, extracting three-dimensional (x, y, z) coordinates and providing 33 key points for each frame[18]. These landmarks played a very important role in capturing and analyzing dynamic movements of sign language gestures, ensuring accurate recognition and interpretation of signs in Bahasa. A visual representation of all 33 landmark points detected by MediaPipe is shown in Fig. 4.

Second, MediaPipe Hand Landmark was adopted to detect key points on the hands, identifying 21 landmarks with x, y, and z coordinates. These coordinates are equally important for accurately capturing hand gestures and movements, ensuring precise recognition of sign language in Bahasa[19]. A visual representation of all 21 landmark points detected by MediaPipe is shown in Fig. 5.

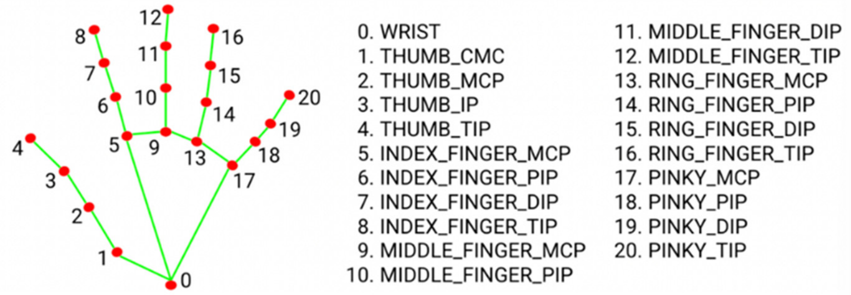


Fig. 5 Landmark Data from MediaPipe Hand Landmarks [19]

The experiments were carried out using three different landmark combinations namely x only, x and y, and x, y, and z. Each video, after being extracted into individual frames, was processed using MediaPipe, which generated a corresponding CSV file. These CSV files contained frame_index, left_hand_landmarks, right_hand_landmarks, and pose_landmarks. It is also essential to elucidate that the experiments were conducted using both raw and normalized landmark data for all three coordinate combinations. As a result, the research produced 18 datasets, derived from the three coordinate sets, three video data variations, and the presence or absence of normalization.

D. Model Experiment and Testing

After obtaining the landmark data, the next step includes model experimentation and testing. The learning process was carried out using three methods namely LSTM, CNN, and a combined CNN-LSTM method, where CNN was applied first, followed by LSTM. The experiment model is presented in Fig. 6.

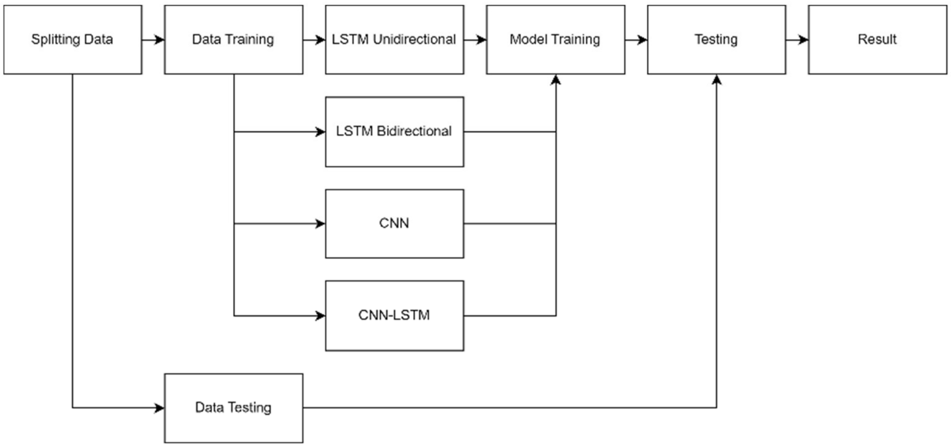


Fig. 6 Model Experiment & Testing

To assess model performance, the results from the three methods namely LSTM, CNN, and CNN-LSTM, were compared using evaluation metrics. This comparison was carried out with the aim of determining the most effective method for recognizing sign language in Bahasa.

E. LSTM

LSTM is an advanced form of Recurrent Neural Networks (RNNs) designed to address the vanishing gradient problem in traditional RNNs. In LSTM, time steps are typically used for classification and its architecture consists of an input layer, hidden layers, and recurrent connections that enable memory retention across extended sequences. LSTM has the capability to store information across more than 1,000 timesteps, making it highly effective for recognizing dynamic sign language gestures. Accordingly, the model operates through a forward pass, where activations are calculated across timesteps from input to output, and a backward pass, where errors are computed at the final timestep and propagated backward for optimization. The structure of the LSTM architecture is shown in Fig. 7.

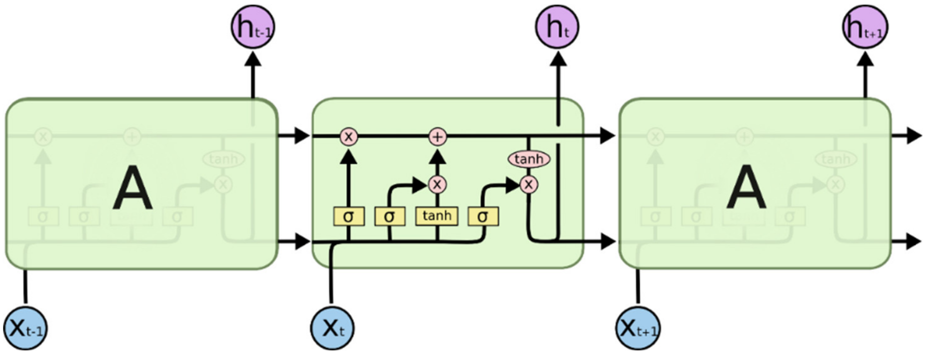


Fig. 7 Long Short-Term Memory (LSTM) Architecture [20]

Based on the figure presented, the LSTM architecture comprises a forget-gate f_t which determines whether or not the input and information from the last timestep (or cell state) will be stored. The sigmoid activation function is typically used in this gate. Following the forget gate, there is an input i_t gate that calculates the input and the cell state with the sigmoid activation to decide whether the value of the cell state will be updated or not. Additionally, there is a tanh layer $\tilde{C}_t$, which is used in the input gate to decide the value of the cell state. If there is an update to the cell state C_t, the forget gate's output will be multiplied by the last cell state and added to the input gate's output and the tanh layer's output. Finally, the output gate o_t, using the sigmoid activation, calculates the result of the cell state with the tanh layer and multiply it with the last output. The architecture comprises a dense layer, which is the neural network layer with the ReLu activation function, and a dropout layer, responsible for ensuring the avoidance of overfitting and the process of learning is faster [21][20].

LSTM Bidirectional is a variation of LSTM with two connected LSTM layers namely forward and backward LSTM layers. The variation typically produces more detailed information, specifically because it considers both previous and future timesteps simultaneously. Based on observation, LSTM Bidirectional learns more than LSTM Unidirectional (or LSTM for short), hence, it improves accuracy[22].

F. CNN

The best-performing dataset from the LSTM model was further evaluated using CNN. These neural networks refine respective predictive accuracy by iteratively adjusting input weights through backpropagation, thereby minimizing prediction loss and enhancing classification performance. The architecture of CNN comprises three principal layers namely the convolutional, pooling, and fully connected layers. The convolutional layer uses kernels to compute dot products with the input, applying activation functions to extract spatial features essential for recognition. Each kernel generates an activation map, which is subsequently stacked along the depth dimension, allowing networks to capture intricate patterns within the data. Following this, the pooling layer reduces data dimensionality by significantly decreasing the number of parameters and computational complexity. This process, which is often executed using max pooling, retains the most prominent features by selecting the highest values within small regions of the activation map. Finally, the fully connected layer establishes connections between every neuron in the previous layer[23].

G. Combining CNN with LSTM

The experiment continued by integrating CNN and LSTM, with CNN applied first, followed by LSTM. In this configuration, CNN extracted spatial features from the input data by processing it through convolutional layers that capture local patterns, followed by pooling layers where dimensionality and computational complexity were reduced. These spatial representations were subsequently fed into the LSTM component, which modeled temporal dependencies by analyzing the sequential structure of the data. By combining CNN's ability to detect spatial features with LSTM's strength in capturing temporal relationships, the model effectively handled spatiotemporal tasks, making it particularly well-suited for dynamic sign language recognition. In accordance with previous research, this hybrid model enhanced both accuracy and adaptability, ensuring better performance in real-world applications[24].

H. Evaluation

The model evaluation process was carried out using the accuracy and loss metrics obtained from TensorFlow, along with a confusion matrix, to assess performance. Accuracy measures the proportion of correctly predicted values compared to actual values, serving as a fundamental classification metric. Typically, higher accuracy signifies a stronger ability to correctly classify sign language gestures[25].

Loss metrics, on the other hand, quantify the model's prediction error during training. This research adopted the use of the 'sparse_categorical_crossentropy' loss function from TensorFlow, which is particularly useful for categorical classification with sparse data. Cross-entropy loss calculates the discrepancy between predicted probabilities and actual labels, assigning penalty values based on deviation. Essentially, larger deviations lead to higher penalties, ensuring the model optimally corresponds its predictions with actual values[26].

Following the cross matrix, a confusion matrix was used to gain a detailed evaluation of classification accuracy. It compares predicted and actual labels, offering insights into correct and incorrect classifications. This tool is particularly useful for identifying patterns of misclassification and refining model performance[27].

During the training process, accuracy and loss metrics were continuously monitored to track the model's learning progress. In the evaluation phase, these metrics were also applied to the testing dataset to ensure an unbiased assessment of model generalization.

III. RESULTS

A. Word Determination

The word determination process identified the 30 most frequently spoken words in the observed videos of religious services. From this initial set, a manual filtering process was carried out, and this reduced the selection to 15 keywords. These words include Amin (Amen), Ayat (Verse), Doa (Pray), Firman (God's Word), Gereja (Church), Haleluya (Hallelujah), Hati (Heart), Ibadah (Worship), Jemaat (Congregation), Kasih (Love), Percaya (Trust), Persembahan (Offering), Roh Kudus (Holy Spirit), Tuhan (God), and Yesus (Jesus). Subsequently, the words were used in the data acquisition phase, where performers were recorded signing each word, all of which were captured in respective videos. This resulted in a total of 315 video files, comprising 21 sets of recordings for the 15 words.

B. Data Preprocessing

The process of capturing video frame-by-frame and extracting landmark data using MediaPipe produced a set of CSV files, with each file corresponding to a single video. In this experiment, MediaPipe was used to generate landmark data using three variations of coordinate representations namely x coordinates only, x and y coordinates, and x, y, and z coordinates. Each CSV file contains columns for the image index, and landmark data from the left hand, right hand, and body. The input length for the model varied based on the number of frames and the selected coordinate dimensions. For x coordinates alone, the dataset included 21 points for the left hand, 21 points for the right hand, and 33 points for the body, totaling 75 points. When including both x and y coordinates, each landmark was represented with two values, increasing the total to 150 points. Finally, when incorporating x, y, and z coordinates, each landmark is represented with three values, bringing the total to 225 points.

The dataset consists of a total of 315 videos, which were classified into 21 sets of videos for 15 words. The total number of videos differed based on the experimental setup as follows.

1) First experiment:

This experiment included the use of 15 video sets, totaling 225 videos, combining four types of shot angles namely eye-level, close-up, low-angle, and wide shots.

2) Second experiment:

Based on the hypothesis that consistent shot angles might yield better performance, this experiment only used eye-level and close-up shots. Additionally, five more sets of videos were added for these shot types, resulting in 14 sets (210 videos).

3) Third experiment:

To test the hypothesis that shot angles might not significantly impact results and using more data could improve model performance, all 315 videos were used, combining eye-level, close-up, low-angle, and wide shots.

These different subsets of videos, 225, 210, and 315, corresponded to the varying experimental conditions and allowed for the examination of the impact of both shot types and dataset size on the performance of the model.

C. Parameter Setting

A consistent set of parameters was used throughout the experiments. These parameters include a sequence length of 110 frames, which was the maximum frame count across all available datasets, meaning some datasets had fewer than 110 frames. Accordingly, the random state used for data splitting was 42, with an 80:20 split ratio of the whole available data for training and testing the developed model respectively. The ratio was selected as it was quite common in machine learning evaluations

D. LSTM

For LSTM, the experiment started with tunning input layer 32, hidden layer 64, batch size, and 1000 epochs. Based on observations, existing LSTM research predominantly applied a dropout of 0.2 along with the Adam optimizer, sparse categorical cross-entropy loss function, and accuracy metrics. The current exploration included the use of 18 different datasets. These include (1) landmark data from 225 videos with only x coordinates. (2) landmark data from 225 videos coordinate x,y. (3) landmark data from 225 videos coordinate x,y,z. (4) landmark data from 225 videos coordinate x normalize. (5) landmark data from 225 videos coordinate x,y normalize. (6) landmark data from 225 videos coordinate x,y,z normalize. (7) landmark data from 210 videos coordinate x. (8) landmark data from 210 videos coordinate x,y. (9) landmark data from 210 videos coordinate x,y,z. (10) landmark data from 210 videos coordinate x normalize. (11) landmark data from 210 videos coordinate x,y normalize. (12) landmark data from 210 videos coordinate x,y,z normalize. (13) landmark data from 315 videos coordinate x. (14) landmark data from 315 videos coordinate x,y. (15) landmark data from 315 videos coordinate x,y,z. (16) landmark data from 315 videos coordinate x normalize. (17) landmark data from 315 videos coordinate x,y normalize. and (18) landmark data from 315 videos coordinate x,y,z normalize. Accordingly, two methods were adopted for the experiments namely LSTM Unidirectional and LSTM Bidirectional. These methods were selected with the primary aim of determining whether LSTM Bidirectional could outperform LSTM unidirectional within the context of the exploration. In total, 36 experiments were conducted in this research. Six of the 36 experiments resulted in an accuracy of more than 90%, as shown in Table 1, all of which were conducted using LSTM Bidirectional.

TABLE 1
THE MOST ACCURATE EXPERIMENT WITH LONG SHORT-TERM MEMORY (LSTM)

Total Video	Model	Coordinate	Train Loss	Train Accuracy	Test Accuracy	Training Time
225	LSTM Bidirectional	Coordinate x normalizes	0.001	100.00%	93.33%	21 m 32 s
225	LSTM Bidirectional	Coordinate x,y,z normalizes	0.000	100.00%	100.00%	21 m 33 s
210	LSTM Bidirectional	Coordinate x,y normalizes	0.000	100.00%	97.62%	19 m 32 s
210	LSTM Bidirectional	Coordinate x,y,z normalizes	0.001	100.00%	92.86%	19 m 30 s
315	LSTM Bidirectional	Coordinate x normalizes	0.001	100.00%	93.33%	19 m 11 s
315	LSTM Bidirectional	Coordinate x,y,z normalizes	0.000	100.00%	100.00%	19 m 32 s

Based on the results presented in the table, it can be seen that LSTM Bidirectional produced the best result compared to LSTM Unidirectional. The outputted information was observed to be richer because the model used considered both previous and future timesteps simultaneously, thereby improving accuracy. Additionally, the landmarks data were normalized, giving room for more accuracy. This is because normalization ensures that no single feature dominates the learning process makes gradients more stable and less prone to exploding or vanishing, ensures the model focuses on learning patterns in the data rather than being distracted by scale differences, can help maintain a stable cell state, and can simplify the initialization of weights and biases in networks. For the dataset consisting of 210 eye-level and close-up shots, the use of normalized x and y coordinates was considered most suitable, as the straightforward nature of the videos did not necessitate the inclusion of the z coordinate, which could introduce unnecessary complexity and potentially confuse the model. However, for the full set of 315 videos, comprising eye-level shots, close-up shots, low-angle shots, and wide shots, using landmark data that included x, y, and z coordinates proved more effective, as it allowed the model to capture and learn from all available spatial features. For this dataset, the use of x coordinates alone was also considered beneficial, as it provided minimal features that were easier for the model to process and learn.

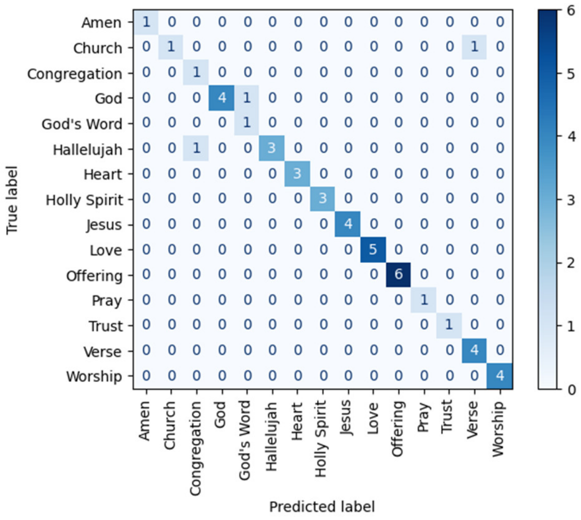
After fine-tuning the model's input layer to 32 units, the hidden layer to 64 units, adjusting the batch size, and running training for 1,000 epochs, the best-performing datasets were further explored through additional tuning. The experimental results, as summarized in Table 2, showed significant improvements. All experiments were conducted using the LSTM Bidirectional model.

TABLE 2
THE TUNNING EXPERIMENT THAT SHOWED IMPROVEMENT

Total Video	Model	Coordinate	Layer	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Training Time
210	LSTM Bidirectional	Coordinate x,y normalize	Input Layer = 32, Layer 1 = 64	16	0.001	100.00%	100.00%	31 m 32 s
210	LSTM Bidirectional	Coordinate x,y normalize	Input Layer = 32, Layer 1 = 64	64	0.009	100.00%	95.56%	12 m 21 s

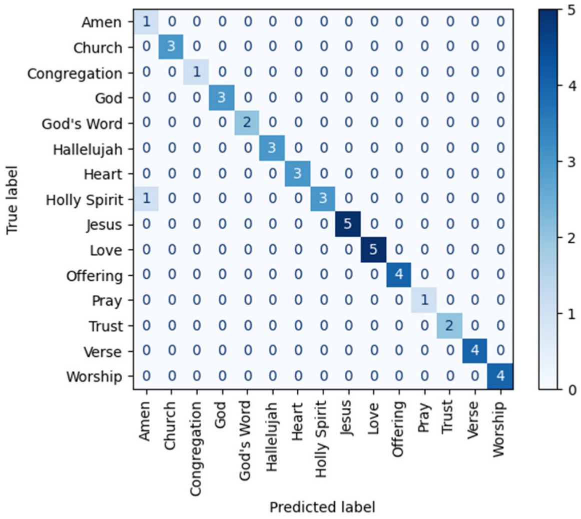
Only the data from the 210 videos using LSTM Bidirectional and normalized x and y coordinates showed significant improvement. The test accuracy was higher when using smaller batch sizes, signifying that the model was able to learn finer details with reduced batch sizes.

The confusion matrix was used to identify words that were difficult to classify. An example of a confusion matrix is shown in Fig. 8, which represents results from 225 videos using LSTM Bidirectional with normalized x coordinates, and Fig. 9, reflecting the results from 210 videos using LSTM Bidirectional with normalized x and y coordinates. The tuning parameters for these experiments included an input layer of 32 units, a hidden layer of 64 units, and a batch size of 32.



	Amen	Church	Congregation	God	God's Word	Hallelujah	Heart	Holly Spirit	Jesus	Love	Offering	Pray	Trust	Verse	Worship
Amen	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Church	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
Congregation	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
God	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0
God's Word	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Hallelujah	0	0	1	0	0	3	0	0	0	0	0	0	0	0	0
Heart	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
Holly Spirit	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
Jesus	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
Love	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
Offering	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0
Pray	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Trust	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Verse	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
Worship	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

Fig. 8 Confusion Matrix 225 Videos Coordinate X



	Amen	Church	Congregation	God	God's Word	Hallelujah	Heart	Holly Spirit	Jesus	Love	Offering	Pray	Trust	Verse	Worship
Amen	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Church	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Congregation	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
God	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
God's Word	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
Hallelujah	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
Heart	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
Holly Spirit	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0
Jesus	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0
Love	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0
Offering	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
Pray	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Trust	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
Verse	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0
Worship	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

Fig. 9 Confusion Matrix 210 Videos Coordinate X, Y

In the confusion matrix of Fig. 8, there is a mistake between God and God's Word because God and God's Word have almost the same movement, which can make the model fail to predict. In another words, Hallelujah and Congregation are likely to have almost the same movement. However, in Church and Verse, failure can appear because of the quality of the performers who are performing the sign language. Like Church and Verse, in the confusion matrix of Fig. 9, there is a mistake predicting Holly Spirit and Amen; these two have very different movements.

E. CNN

The results showed that LSTM Bidirectional with x, y, and z coordinates provided the most accurate predictions. Based on this result, the CNN model was tested using the same dataset across three variations of video sets comprising 225, 210, and 315 videos respectively. The tunable parameters in CNN included the convolutional layer configuration, kernel size, pooling size, dense layer setup, and batch size, while the number of epochs was fixed at 1000. In the CNN experiments, Conv1D layers were used with the ReLU activation function, a dropout rate of 0.2, and dense layers, which included the use of ReLU for the first and softmax for the final classification layer. The model was trained using the Adam optimizer, with sparse categorical cross-entropy as the loss function and accuracy as the evaluation metric. The detailed performance outcomes of these configurations are presented in Table 3.

TABLE 3
THE RESULT FROM THE CONVOLUTIONAL NEURAL NETWORK (CNN) MODEL

Total Video	Convolutional Layer	Kernel Size	Pool Size	Dense Layer	Batch Size	Train Loss	Train Accuracy	Test Accuracy	Training Time
225	256	1	1	512	32	0.384	95.56%	86.67%	32 m 23 s
210	256	1	1	512	32	0.125	97.62%	73.81%	34 m 24 s
315	256	1	1	512	32	0.384	95.56%	86.67%	36 m 43 s

The best results were obtained when the convolutional layer was set to 256 and the dense layer to 512. Additionally, using a kernel size of 1 and a pool size of 1 produced optimal performance. Based on the observations made, batch size did not significantly impact the accuracy of the model during training and testing. This was attributed to the fact that increasing the number of units in both the convolutional and dense layers enhanced the model's capacity to learn intricate patterns and develop deeper representations of the data. As a result, the model became more adept at capturing finer and more abstract features in sign language recognition. It is also important to state that adopting a kernel size of 1 and a pool size of 1 made the model potentially more sensitive to subtle details.

F. CNN-LSTM

The experiment proceeded by integrating the CNN and LSTM methods, with CNN applied first, followed by LSTM. The best tuning parameters from the previously developed models were adopted. The CNN model was configured with a convolutional layer of 256, a kernel size of 1, a pooling size of 1, and a dense layer of 512. Meanwhile, the LSTM Bidirectional model was optimized with an input layer of 32 and a hidden layer of 64. A batch size of 32 was also used, and the dataset comprised 225 videos with normalized x, y, z coordinates, 210 videos with normalized x, y, z coordinates, and 315 videos with normalized x, y, z coordinates. The experimental results are presented in Table 4.

Compared to the standalone CNN method, the combination of CNN with LSTM Bidirectional showed improved training and testing accuracy. This enhancement was attributed to the complementary strengths of both models, with CNN effectively extracting spatial features from sequential data, and LSTM Bidirectional capturing both long-term and short-term dependencies.

TABLE 4
THE RESULT FROM THE CNN-LSTM MODEL

Total Video	Train Loss	Train Accuracy	Test Accuracy	Training Time
225	0.126	96.67%	88.89%	27 m 33 s
210	0.046	98.21%	95.24%	26 m 32 s
315	0.126	96.67%	88.89%	28 m 37 s

IV. DISCUSSION

The models used in this research were designed particularly to recognize 15 gestures allocated to specific words namely Amen, Verse, Pray, God's Word, Church, Hallelujah, Heart, Worship, Congregation, Love, Trust, Offering, Holy Spirit, God, and Jesus. These models showed promising effectiveness and could be potentially applied to other dynamic sign language recognition tasks in Bahasa. The experimental results signified the presence of a strong correlation between video recording techniques and the effectiveness of the landmark data obtained using MediaPipe. The obtained results showed that videos captured with straightforward shots, such as eye-level or close-up angles, using only the x and y coordinates were sufficient, as the z coordinate remained relatively constant and did not interfere with the learning process. However, for videos taken from multiple angles, incorporating the z coordinate was considered very important for accurate recognition. Using only the x coordinate was also to be a viable option due to its simplicity.

Based on the observations made, normalizing the landmark data significantly enhanced model accuracy by ensuring that no single feature dominated the learning process. This method helped to stabilize the gradients, prevented issues related to exploding or vanishing gradients, and enabled the model to focus on recognizing patterns rather than being influenced by scale differences. Additionally, normalization facilitated a stable cell state while simplifying weight and bias initialization.

Among the models tested, LSTM Bidirectional achieved the highest performance, with test accuracy reaching 100%. This remarkable accuracy was attributed to the model's ability to process temporal information in both forward and backward directions, providing a richer context for learning. However, the CNN model attained a maximum test accuracy of 86.67%, and the CNN-LSTM hybrid model showed an improvement with a peak accuracy of 95.24%. Regardless of the fact that the hybrid model benefited from combining spatial feature extraction with temporal processing, it still fell short of the accuracy achieved by LSTM Bidirectional.

The results signified that while CNN was effective for extracting spatial features, its performance in dynamic sign language recognition remained limited without the temporal processing capabilities of LSTM. The ability of the LSTM Bidirectional model to interpret gestures in a contextual flow makes it significantly more effective for this task. Future research should focus on enhancing the model to recognize complete sentences in sign language and explore advanced sequence modeling methods to further improve accuracy and practical applicability.

The current research is the first of its kind in several ways. First, it focuses on Bahasa rather than English sign languages [7] [28], recognizing the unique characteristics of Bahasa sign language. Second, the exploration specifically identified symbolic signs instead of alphabetical [8] [9] [11] [12] [13] [14] [28]or numerical signs[10]. Third, it adopted the use of simple video cameras rather than specialized devices such as Microsoft Kinect Xbox [29] [15], making sign language recognition more accessible. Fourth, the advancements in this research surpass previous investigations, such as those by Anggita, Khotimah, and Suciati[15], which relied on Kinect for data collection and recognition. The methodology used enabled video acquisition using widely available devices, including standard or smartphone cameras, thereby broadening accessibility. By integrating LSTM and CNN, the detection of dynamic symbolic signs in Bahasa was enhanced. However, the current model is limited to recognizing sign language on a word-by-word basis. Future research should focus on expanding the developed model's capabilities to recognize and interpret entire sentences, thereby further advancing its potential for real-world applications.

The current research has several limitations, including the fact that, first, the recognized vocabulary was restricted to 15 symbolic words related to religious activities. Expanding the word set could enhance the applicability of the model. Second, model evaluation was based on accuracy and loss metrics. While these metrics were used to effectively measure the performance of the model, incorporating additional evaluation methods in future research could provide a more comprehensive assessment. Third, regardless of the fact that the dataset used was sufficient for this investigation, incorporating a larger and more diverse dataset could further strengthen the results and improve model generalization.

V. Conclusions

In conclusion, using the models developed during the course of this research, fifteen words related to religious activities were successfully identified, forming the basis for recognizing Indonesian sign language. The best results were achieved with an LSTM Bidirectional model, using video data captured from various angles and normalized x, y, and z coordinates extracted through MediaPipe. Accordingly, the LSTM model achieved a testing accuracy ranging from 92.86% to 100% and CNN-based recognition produced lower accuracy, between 73.81% and 86.67%. The CNN-LSTM hybrid model, on the other hand, performed better than CNN alone but remained below the LSTM Bidirectional, with testing accuracy ranging from 88.89% to 95.24%.

For future research, this model could be expanded to recognize a broader vocabulary of Indonesian sign language across different contexts. Furthermore, novel explorations on the capabilities of MediaPipe for data extraction could enhance the accuracy of landmark detection. CNN could also be leveraged for frame-by-frame image extraction by embedding skeletal representations with MediaPipe before inputting into CNN for spatial learning, followed by LSTM for sequential analysis. Lastly, comparing the developed model with other existing models and incorporating additional evaluation metrics, such as precision, recall, and F1-score, would provide a more comprehensive assessment of its performance.

Author Contributions: *Ivana Valentina Lemmuela*: Conceptualization, Data Curation, Formal Analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing - Original Draft. *Mewati Ayub*: Conceptualization, Data Curation, Investigation, Methodology, Supervision, Writing – Review & Editing. *Oscar Karnalim:* Conceptualization, Data Curation, Investigation, Methodology, Supervision, Writing – Review & Editing.

All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Maranatha Christian University.

Conflicts of Interest: The authors declare no conflict of interest.

Data Availability: The data is collected manually by shooting the performers who performed the sign language. It is available at https://github.com/oscarkarnalim/signlanguagerecogbahasa

Informed Consent: Informed Consent was obtained, and a detailed explanation was presented in the Methods section.

Institutional Review Board Statement: Not applicable.

Animal Subjects: There were no animal subjects.

ORCID:
Ivana Valentina Lemmuela: https://orcid.org/0009-0000-0055-8424
Mewati Ayub: https://orcid.org/0000-0003-2584-4317
Oscar Karnalim: https://orcid.org/0000-0003-4930-6249

References

[1] N. P. Kirana, N. D. Iroth, and N. C. Salsabila, “Fenomena Penggunaan Bahasa Isyarat Bagi Penyandang Tuna Rungu di Sekolah Inklusi,” *Hasanuddin Journal of Sociology (HJS)*, vol. 4, no. 2, pp. 119–134, 2022.

[2] R. A. Mursita, “RESPON TUNARUNGU TERHADAP PENGGUNAAN SISTEM BAHASA ISYARAT INDONESA (SIBI) DAN BAHASA ISYARAT INDONESIA (BISINDO) DALAM KOMUNIKASI,” *INKLUSI*, vol. 2, no. 2, p. 221, Dec. 2015, doi: 10.14421/ijds.2202.

[3] A. Rahagiyanto, “IDENTIFIKASI EKSTRAKSI FITUR UNTUK GERAKAN TANGAN DALAM BAHASA ISYARAT (SIBI) MENGGUNAKAN SENSOR MYO ARMBAND,” *Jurnal Matrik*.

[4] M. E. Al Rivan, H. Irsyad, K. Kevin, and A. T. Narta, “Pengenalan Alfabet American Sign Language Menggunakan K-Nearest Neighbors Dengan Ekstraksi Fitur Histogram Of Oriented Gradients,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 5, no. 3, Jan. 2020, doi: 10.28932/jutisi.v5i3.1936.

[5] M. E. Al Rivan and S. Hartoyo, “Klasifikasi Isyarat Bahasa Indonesia Menggunakan Metode Convolutional Neural Network,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 2, Aug. 2022, doi: 10.28932/jutisi.v8i2.4863.

[6] Y. Pratama, E. Marbun, Y. Parapat, and A. Manullang, “Deep convolutional neural network for hand sign language recognition using model E,” *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 1873–1881, Oct. 2020, doi: 10.11591/eei.v9i5.2027.

[7] R. Mohamed Abdulhamied, M. M. Nasr, and S. N. Abdul Kader, “Real-time recognition of American sign language using long-short term memory neural network and hand detection,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 1, p. 545, Apr. 2023, doi: 10.11591/ijeecs.v30.i1.pp545-556.

[8] S. Supria, D. Herumurti, and W. N. Khotimah, “PENGENALAN SISTEM ISYARAT BAHASA INDONESIA MENGGUNAKAN KOMBINASI FITUR STATIS DAN FITUR DINAMIS LMC BERBASIS L-GCNN,” *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 14, no. 2, p. 217, Jul. 2016, doi: 10.12962/j24068535.v14i2.a574.

[9] A. A. Gafar and J. Y. Sari, “Sistem Pengenalan Bahasa Isyarat Indonesia dengan Menggunakan Metode Fuzzy K-Nearest Neighbor,” *Jurnal ULTIMATICS*, vol. 9, no. 2, pp. 122–128, Apr. 2018, doi: 10.31937/ti.v9i2.671.

[10] I. Mahfudi, M. Sarosa, R. Andrie Asmara, and M. Azrino Gustalika, “Indonesian Sign Language Number Recognition using SIFT Algorithm,” *IOP Conf Ser Mater Sci Eng*, vol. 336, p. 012010, Apr. 2018, doi: 10.1088/1757-899X/336/1/012010.

[11] O. D. Nurhayati, D. Eridani, and M. H. Tsalavin, “Sistem Isyarat Bahasa Indonesia (SIBI) Metode Convolutional Neural Network Sequential secara Real Time,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 4, pp. 819–828, Aug. 2022, doi: 10.25126/jtiik.2022944787.

[12] M. Sholawati, K. Auliasari, and FX. Ariwibisono, “PENGEMBANGAN APLIKASI PENGENALAN BAHASA ISYARAT ABJAD SIBI MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN),” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 1, pp. 134–144, Mar. 2022, doi: 10.36040/jati.v6i1.4507.

[13] M. E. Al Rivan and S. Hartoyo, “Klasifikasi Isyarat Bahasa Indonesia Menggunakan Metode Convolutional Neural Network,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 2, Aug. 2022, doi: 10.28932/jutisi.v8i2.4863.

[14] I. J. Thira, D. Riana, A. N. Ilhami, B. R. S. Dwinanda, and H. Choerunisya, "Pengenalan Alfabet Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Convolutional Neural Network," *Jurnal Algoritma*, vol. 20, no. 2, pp. 421–432, Oct. 2023, doi: 10.33364/algoritma/v.20-2.1480.

[15] T. Anggita, W. N. Khotimah, and N. Suciati, "Pengenalan Bahasa Isyarat Indonesia dengan Metode Dynamic Time Warping (DTW) Menggunakan Kinect 2.0," *Jurnal Teknik ITS*, 2018.

[16] H. R. Al Fajri, J. Jayanta, and B. T. Wahyono, "Sistem Pengenalan Gerak Bahasa Isyarat Dengan Colored Motion History Image dan Convolutional Neural Network," in *Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)*, Fakultas Ilmu Komputer Universitas Pembangunan Nasional Veteran Jawa Timur, Aug. 2022.

[17] Y. Effendi, Y. Kristian, L. Z. P.C.S.W, and H. Yutanto, "Pemanfaatan Mediapipe Body Pose Estimation dan Dynamic Time Warping untuk Pembelajaran Tari Remo," *Jurnal Teknologi dan Manajemen Informatika*, vol. 9, no. 2, pp. 183–190, Dec. 2023, doi: 10.26905/jtmi.v9i2.10408.

[18] G. Kaur, G. Jaju, D. Agarwal, K. Iyer, and C. M. Prashanth, "Implementation of Geriatric Agility Detection Using MediaPipe Pose," *International Journal of Recent Advances Multidisciplinary Topics*, vol. 3, no. 6, Jun. 2022.

[19] K. Kavana and N. Suma, "RECOGNIZATION OF HAND GESTURES USING MEDIAPIPE HANDS," *International Research Journal of Modernization in Engineering Technology and Science* , vol. 4, Jun. 2022.

[20] B. A. H. Kholifatullah and A. Prihanto, "Penerapan Metode Long Short Term Memory Untuk Klasifikasi Pada Hate Speech," *Journal of Informatics and Computer Science (JINACS)*, pp. 292–297, Jan. 2023, doi: 10.26740/jinacs.v4n03.p292-297.

[21] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, Oct. 2019, doi: 10.2478/jaiscr-2019-0006.

[22] A. R. Isnain, A. Sihabuddin, and Y. Suyanto, "Bidirectional Long Short Term Memory Method and Word2vec Extraction Approach for Hate Speech Detection," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 14, no. 2, p. 169, Apr. 2020, doi: 10.22146/ijccs.51743.

[23] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[24] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, Apr. 2015, pp. 4580–4584. doi: 10.1109/ICASSP.2015.7178838.

[25] W. M. Baihaqi and A. Munandar, "Sentiment Analysis of Student Comment on the College Performance Evaluation Questionnaire Using Naïve Bayes and IndoBERT," *JUITA : Jurnal Informatika*, vol. 11, no. 2, p. 213, Nov. 2023, doi: 10.30595/juita.v11i2.17336.

[26] B. N. Chaithanya, T. J. Swasthika Jain, A. U. Ruby, and A. Parveen, "An approach to categorize chest X-ray images using sparse categorical cross entropy," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 3, p. 1700, Dec. 2021, doi: 10.11591/ijeecs.v24.i3.pp1700-1710.

[27] M. N. A. Saputro, F. Liantoni, and D. Maryono, "Application of Convolutional Neural Network Using TensorFlow as a Learning Medium for Spice Classification," *Ultimatics : Jurnal Teknik Informatika*, pp. 8–15, Jul. 2024, doi: 10.31937/ti.v16i1.3304.

[28] I. Fareza, R. Busdin, M. E. Al Rivan, and H. Irsyad, "Pengenalan Alfabet Bahasa Isyarat Amerika Menggunakan Edge Oriented Histogram dan Image Matching," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 4, no. 1, Apr. 2018, doi: 10.28932/jutisi.v4i1.747.

[29] T. Handhika, R. I. M. Zen, Murni, D. P. Lestari, and I. Sari, "Gesture recognition for Indonesian Sign Language (BISINDO)," *J Phys Conf Ser*, vol. 1028, p. 012173, Jun. 2018, doi: 10.1088/1742-6596/1028/1/012173.

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