Sentence-Level Indonesian Sign Language (BISINDO) Recognition Using 3D CNN-LSTM and 3D CNN-BiLSTM Models

Katriel Larissa Wiguna, Rojali
Computer Science Department-BINUS Graduate Program–Master of Computer Science,
Bina Nusantara University, Jakarta, Indonesia, 11480

***Abstract*—Sign Language Recognition (SLR) has been an active area of research, but sentence-level SLR remains relatively underexplored. While most studies focus on recognizing individual signs, understanding full sentences presents greater challenges. This research proposes a sentence-level SLR using a combination of 3D Convolutional Neural Networks (3D CNN) for spatio-temporal feature extraction with sequential modeling using Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM). Connectionist Temporal Classification (CTC) is also used to enable training without word-level annotations. In this study, we used the Indonesian Sign Language (BISINDO) dataset, specifically the DKI Jakarta version, consisting of 900 videos representing 30 sentences, which was expanded to 3600 videos through data augmentation techniques such as speed variation and brightness adjustments. All videos underwent preprocessing to ensure data quality, and Bayesian Optimization was applied for hyperparameter tuning to obtain optimal configurations for each model. Both models were trained with CTC loss and evaluated using Word Error Rate (WER). The 3DCNN-LSTM model achieved a WER result of 59.21%, while the 3DCNN-BiLSTM presents a significantly better performance with a WER of 2.77%. Despite these promising results, the models' ability to generalize across different signers may require further research, as the dataset used in this research involved only a single signer.**

Keywords*—Sign Language Recognition; BISINDO (Indonesian Sign Language); 3D Convolutional Neural Network (3D CNN); Long Short-Term Memory (LSTM) Network; Bidirectional Long Short-Term Memory (BiLSTM); Connectionist Temporal Classification (CTC)

I. INTRODUCTION

As social creatures, communication is an essential part that is inseparable from human life. Through communication, humans are able to exchange ideas, build relationships, and collaborate to fulfill their needs and achieve their goals. Broadly speaking, communication can be classified into verbal communication, which uses spoken or written words, and non-verbal communication, which relies on facial expressions, gestures, and body language.

Despite the fact that verbal communication, especially speaking, is the most common way humans interact, it is undeniable that some people cannot speak. Deaf individuals, unlike the general population, are among those who are unable to speak. For this reason, the deaf community communicates in their own, unique way, which is through the use of sign language.

In Indonesia, there are two types of sign language used: Indonesian Sign Language System (SIBI) and Indonesian Sign Language (BISINDO). SIBI is a sign language system that follows standard Indonesian grammar, which includes basic signs that symbolizes words, and additional signs for prefixes, suffixes, and particles. However, it is not fully accepted by the deaf community, who find it difficult for daily use. Although SIBI is used in formal education, BISINDO is more commonly used by the deaf for everyday communication. Research at PGMI UIN Sunan Kalijaga shows BISINDO is easier to understand, more natural, expressive, and effective than SIBI [1].

The difference in communication styles between the deaf and hearing communities creates a communication gap, which can lead to social distance and discrimination within society. Therefore, a system is needed to help bridge this gap. As a solution to this gap, sign language recognition (SLR) systems have been widely developed [2]. The goal of SLR is to develop a method capable of detecting and understanding the sign language gestures made by individuals [2]. Various models have been applied in the development of sign language recognition systems, ranging from Convolutional Neural Networks (CNN) [3], Bidirectional Recurrent Neural Networks [4], Transformers [5], Long Short-Term Memory Networks (LSTM) [6], deep convolutional networks [7], to a combination of 2 or more models [8].

Considering the fact that Indonesia is one of the four countries in Asia with a fairly high prevalence of hearing loss at 4.6%, and a deafness prevalence of 0.4% across all age groups in seven provinces, a continuous sign language recognition, also known as sentence-level SLR, is very feasible to be developed to help deaf individuals communicate with hearing individuals. While this system helps translate what deaf individuals want to convey to hearing individuals, it also serves as a tool for hearing individuals to learn sign language. In Indonesia itself, there has been a research on sentence-level SLR using 3D CNN and Bidirectional Recurrent Neural Networks (BiRNN) [4]. However, the model developed utilizes the SIBI dataset, which is not frequently used by deaf individuals in their daily interactions. Besides, it shows a relatively high word error rates (WER) and character error rates (CER), which may be caused by a mismatch between the model and the dataset used, such as the presence of noise in the dataset due to suboptimal preprocessing. Additionally, sentence-level SLR is generally more challenging than alphabet or word-level recognition due to the need to capture both manual and non-manual features, while also modeling temporal dependencies across multiple gestures and handling variations in gesture sequences.

In response to this problem, this study proposes a sentence-level SLR model for Indonesian Sign Language (BISINDO) using 3D CNN-LSTM and 3D CNN-BiLSTM models. To address limitations in previous studies, the dataset consists of BISINDO videos collected with consistent lighting, framing, and other conditions to minimize noise. LSTM and BiLSTM are used instead of BiRNN due to their gating mechanisms, which better handle vanishing gradients and capture long-term dependencies in gesture sequences [9].

Since BISINDO naturally develops within the Deaf community, there are lots of variations in BISINDO gestures across different regions, reflecting the local culture in each area. This results in several regional variants of BISINDO, such as BISINDO DKI Jakarta, BISINDO West Java, BISINDO East Java, and others. However, since DKI Jakarta is the province with the highest population density in Indonesia [10] and as a central area where many communities interact, the DKI Jakarta version of BISINDO is likely to be more widely used and recognized than variants of other regions. Therefore, we chose to focus on BISINDO DKI Jakarta in this research.

To sum up, this study aims to develop and evaluate a sentence-level BISINDO DKI Jakarta recognition using a combination 3D CNN and LSTM model, as well as 3D CNN and BiLSTM model. While previous studies have focused on word or alphabet-level recognition, only a few have focused on sentence-level SLR. This gap motivates the development of a model capable of recognizing sentence-level sign language gestures. The proposed model is expected to provide a basis for developing future systems that support communication between Deaf and hearing communities. It has the potential to support sign language learning, contribute to the development of applications such as sign language translators, and enable future innovations in automated public service systems.

To address these objectives, this study seeks to answer the following research questions: 1) How can hybrid 3D CNN-LSTM and 3D CNN-BiLSTM models be developed to effectively recognize sentence-level BISINDO gestures? 2) How do the performance of the 3D CNN-LSTM and 3D CNN-BiLSTM models compare in terms of their accuracy in recognizing word sequences within BISINDO sentences?

The next section of this paper is organized as follows: Section II presents related works on Sign Language Recognition, Section III presents the methods used in this research. The results and discussion are presented in Section IV and Section V, respectively. Finally, Section VI presents the conclusion and future works of this research.

II. Related Works

To date, there has been considerable research focused on sign language recognition systems. Different algorithms have been implemented by previous researchers to develop a sign language recognition system that can perform with good accuracy.

In 2018, Ariesta et al. proposed a sentence-level Indonesian sign language recognition system using 3D CNN and Bidirectional Recurrent Neural Networks. The study utilized SIBI dataset, which includes 30 sentences in SIBI. The researchers developed several models for this system, all combining 3D CNN and bidirectional RNN. However, the evaluation revealed that the performance of the developed models was not optimal, as indicated by the high Word Error Rate (WER) of 85%-90% and Character Error Rate (CER) of 65%-77% [4]. This study is notable for attempting sentence-level sign language recognition, which is a challenging task. The use of 3D CNN-BiRNN is also suitable for capturing spatio-temporal features. However, the dataset and the model seem to be mismatched, resulting in suboptimal performance. This highlights the importance of preparing the dataset carefully for sentence-level recognition.

Aljabar and Suharjito in 2020 proposed a word-level sign language recognition system, utilizing BISINDO dataset consisting of two alphabets and eight words in BISINDO. They developed CNN, LSTM, and a combination of CNN and LSTM model, in which the CNN model achieves an accuracy of 73%, LSTM model achieves an accuracy of 81%, and the combination of CNN and LSTM model achieves the highest accuracy, reaching 90% [11].

Another research in 2020 by Latif et al. proposed an alphabet-level sign language recognition system to detect alphabet in Arabic sign language. They used a deep convolutional network, which was trained and evaluated using a dataset consisting of 50000 images of Arabic signs. After evaluation, it is found that this model achieves an accuracy of 97.6% [7]. While this study achieves a high accuracy in recognizing Arabic letters with its large and diverse dataset, it is still limited to letters rather than word or sentence-level recognition.

In the same year, a continuous sign language neural machine translation was proposed by a group of researchers from China. In this research, they used a dataset consisting of 50 sentences in Chinese Sign Language. The model proposed in this research is the ST-LSTM fusion attention network, which is then called Bi-ST-LSTM-A. This model achieves an accuracy of 81.22% on the CSL dataset, 76.12% on the RWTH-PHOENIX-Weather dataset, and 75.32% on the RWTH-PHOENIX-2014T dataset [8].

In 2021, a group of researchers from Saudi Arabia proposed an Arabic Sign Language recognition to detect alphabets in Arabic Sign Language using R-CNN, in which they achieved an accuracy of 93% based on the evaluation conducted [12]. Although this study focuses only on recognizing alphabets in Arabic Sign Language, the model proposed in this study can detect hands and recognize gestures with a high accuracy, even in a complicated background.

Another researcher from Indonesia also proposed a sign language detection system for BISINDO in 2022. They used a dataset of nine class, which consists of the alphabet A-E, and three words which are “saya”, “kamu”, and “I love you”. They used CNN to detect those classes, in which they achieved an accuracy of 99.82% [13]. This study still needs a lot of improvements, especially in terms of the dataset size, as the limited vocabulary makes it unreliable in real-world scenarios.

Another alphabet-level sign language recognition is also proposed by Murali et al. in 2022. They used a CNN model to detect alphabets in American Sign Language, in which they got an accuracy of 98% [3].

In the same year, research by Kothadiya et al. proposed a system to detect words in Indian Sign Language. They used deep learning models, especially the Long Short-Term Memory (LSTM) and GRU. The model, which consists of 1 LSTM layer and GRU, achieves an accuracy of 97% for more than 11 word classes [14]. This shows a high accuracy for Indian Sign Language Recognition in a challenging and uncontrolled environment, representing natural conditions. However, the dataset is still limited in diversity, containing only 11 words, which may restrict the model’s ability to recognize sign language in real-world applications.

Another alphabet and word-level sign language recognition is proposed by Shin et al. in 2023. They combined a CNN and a transformer model to create a model to detect alphabet and words in the KSL Dataset. After evaluation, almost all alphabets are identified correctly, two alphabets achieved an accuracy of 95%, and two others gained an accuracy of 84%-90%. Testing was also done on the proposed dataset, which consists of 20 classes, achieving a score of 98.5% for precision, 98.35% for recall, 98.4% for F1-score, and 98.3% for accuracy [5]. This study effectively combines CNN and transformer architectures, resulting in high accuracy with relatively low computational cost. However, it is limited to word-level recognition and relies on a relatively small dataset, which limits its applicability to continuous, real-world sign language scenarios.

In 2023, a group of researchers from India proposed a system using the LSTM networks to detect words in Indian Sign Language. This model achieves an accuracy of 87% [6]. This study achieves a relatively high accuracy, and it is strong in its practical applicability, demonstrating a usable, real-time sign-action recognition system. However, it is limited to word-level recognition and does not address sentence-level sequences.

Lastly, in 2024, a hybrid deep learning model was proposed for a real-time Arabic Sign Language. This study presents a new, custom dataset containing 10 static gesture words and 10 dynamic gesture words. CNN and LSTM were used in this study, with the CNN achieving accuracy rates of 94.40% and the LSTM achieving 82.70% [15]. The results highlight the strong potential of the proposed model, especially proving CNN’s effectiveness in recognizing static signs. However, the dataset used in this research is relatively small, which makes it less applicable in real-world scenarios. Expanding the dataset in the future, especially for dynamic gestures, is essential to achieve a more reliable sign language recognition system.

From the previous research, it can be concluded that sign language recognition systems with various models, especially CNN and LSTM can be used to detect gestures in sign language with a fairly good level of accuracy, with an average accuracy of more than 90% for word-level sign language recognition. However, research related to sentence-level sign language recognition is still limited. Based on previous research that has been discussed, there are only two groups of researchers who studied sentence-level sign language recognition, where 1 of them shows a large error rate, and 1 the other shows fairly good accuracy results (75%- 82%), although not as good as the accuracy of the system in the word-level sign language recognition. Therefore, referring to previous research, this study will develop sentence-level sign language recognition using a combination of 3D CNN and LSTM models. This study was conducted to detect gestures in the DKI Jakarta version of BISINDO.

III. Methodology

This section presents the detailed research workflow, describing the step-by-step process involved in preparing the data, building and training the model, to evaluating its performance for BISINDO DKI Jakarta sentence-level SLR.

A. Problem Identification

In the problem identification phase, we focused on highlighting specific issues within the field of Sign Language Recognition (SLR), aiming to address gaps in current research. This process was done through an extensive review of recent studies and literature related to SLR. Even though numerous researchers have successfully developed SLR, most of them still focus on word-level Sign Language Recognition, so the communication gap between deaf and hearing individuals is not fully addressed.

In order to address the problem, this research focuses on developing a sentence-level SLR using the BISINDO dataset, a sign language which is preferred by deaf individuals in Indonesia. While the BISINDO dataset varies across Indonesia, this research focuses particularly on the DKI Jakarta version of BISINDO.

B. Video Acquisition

While several BISINDO datasets are available at the word or alphabet level, no publicly accessible dataset exists for sentence-level BISINDO (especially the DKI Jakarta version). Due to this gap, we created our own BISINDO DKI Jakarta dataset to be used in this research.

The video acquisition phase began with compiling a list of sentences to be recorded, which will then be used in this study. To construct the sentences, we utilizes vocabularies which are included in the 50 high frequency words based on research conducted by Siagian, which are recommended as mandatory vocabularies to be mastered when learning a new foreign language [16]. From this list, we selected several words to construct the sentences. The selection of these words was based on their frequency in everyday conversations and their applicability within Bahasa Isyarat Indonesia (BISINDO). While most words were selected from the 50 high-frequency words list, additional words outside the list were also incorporated as needed to form meaningful sentences. The sentences constructed from these words are then recorded to be used as input for training and testing our sentence-level sign language recognition model. The list of sentences which will be included in this dataset can be seen in Table I. The “Sentence” column contains the original sentence in Indonesian, while the “Sign Gloss” column represents the translation of the sentence into BISINDO gloss notation. As shown in the examples, the gloss often differs from the spoken sentence structure, illustrating the linguistic differences between spoken Indonesian and BISINDO, which the recognition model must learn to accurately align sentences with their corresponding signs.

TABLE I. LIST OF SENTENCES IN THE DATASET

No.	Sentence	Sign Gloss
1.	Aku bangun jam sepuluh.	AKU BANGUN JAM SEPULUH
2.	Aku tidur jam sepuluh malam sampai jam lima pagi.	AKU MALAM TIDUR JAM SEPULUH SAMPAI PAGI JAM LIMA
3.	Aku mau makan.	AKU MAU MAKAN
4.	Aku punya kakak.	AKU PUNYA KAKAK
5.	Aku suka kamu.	AKU SUKA KAMU
6.	Di mana Bapak dan Ibu?	BAPAK DAN IBU DIMANA
7.	Halo, nama aku Katriel	HALO NAMA AKU FS:KATRIEL
8.	Ibu aku punya kucing dan ikan.	IBU AKU PUNYA KUCING DAN IKAN
9.	Kakak aku suka makan.	KAKAK AKU SUKA MAKAN
10.	Berapa anak yang kamu punya?	KAMU ANAK BERAPA
11.	Jam berapa kamu bangun?	KAMU BANGUN JAM BERAPA
12.	Apakah kamu sudah makan?	KAMU MAKAN SUDAH
13.	Keluarga aku terdiri dari lima orang.	KELUARGA AKU ORANG LIMA
14.	Berapa jumlah anggota keluarga kamu?	KELUARGA KAMU ORANG BERAPA
15.	Kucing makan apa?	KUCING MAKAN APA
16.	Kucing makan ikan.	KUCING MAKAN IKAN
17.	Siapa nama dia?	NAMA DIA SIAPA
18.	Siapa nama kamu?	NAMA KAMU SIAPA
19.	Nama isyarat kamu apa?	NAMA PANGGIL KAMU APA
20.	Rumah aku di Jakarta	RUMAH AKU FS:JAKARTA
21.	Rumah aku nomor lima.	RUMAH AKU NOMOR LIMA
22.	Rumah kamu di mana?	RUMAH KAMU DIMANA
23.	Rumah kamu nomor berapa?	RUMAH KAMU NOMOR BERAPA
24.	Sekarang jam berapa?	SEKARANG JAM BERAPA
25.	Selamat hari Ibu.	SELAMAT HARI IBU
26.	Selamat hari kemerdekaan Indonesia.	SELAMAT HARI KEMERDEKAAN INDONESIA
27.	Selamat malam, aku mau tidur.	SELAMAT MALAM AKU MAU TIDUR
28.	Selamat pagi, Bapak dan Ibu.	SELAMAT PAGI BAPAK DAN IBU
29.	Terima kasih	TERIMAKASIH
30.	Umur kamu berapa?	UMUR KAMU BERAPA

The recordings were carried out by a single participant – the author of this research – who is a non-native signer but has completed a sign language course, which provides sufficient experience to ensure accurate movements and expressions for the dataset. The decision to record with one signer was made due to limited resources, which made it challenging to involve multiple native signers. However, the recordings were carefully done to maintain consistency and clarity.

In total, 30 BISINDO sentences (DKI Jakarta version) were recorded, each repeated 30 times, using a smartphone camera at 30 fps. The video recording was carried out under controlled conditions, such as:

- Videos were recorded with a plain white background to minimize unnecessary visual distractions, to help the model focus on the signer.
- Videos were recorded indoors under sufficient lighting, ensuring every gesture was clearly visible.
- The camera was positioned to capture the signer’s upper body (from head to waist), so that both hand movements and relevant facial expressions were clearly captured.

Nevertheless, relying on a single non-native signer remains a key challenge of this dataset, as it may limit the model’s ability to generalize across different signers. To partially address this challenge, data augmentation techniques were applied to enhance the variability of the dataset, which will be described in the following section.

C. Video Preprocessing

To ensure that the recorded videos could be effectively used for model training, we applied several preprocessing steps. Firstly, we cropped the recorded videos to remove unnecessary parts of the frame, allowing the model to better focus on the signer’s gestures and expressions. It is ensured that all videos have a resolution of 1080x1080 pixels. Next, the video frame rate was reduced from 30 fps to 15 fps. This was done to reduce computational load without changing the duration of the video. By reducing the fps, the total number of frames in each video can be cut in half while preserving necessary temporal information. Then, each frames are resized from 1080x1080 pixels to 256x256 pixels to maintain consistency in the input data, while also improving computational efficiency without losing essential details such as hand and facial features.

Next, padding was applied to the videos to ensure that the number of frames across all videos is the same. Since this research is focused on sentence-level sign language recognition, the number of frames of each video in the dataset will be different, due to the difference in sentence length or the variability in signing speed. Hence, padding was done by determining the maximum number of frames in a video across all videos in the dataset. The videos with fewer frames than the maximum number of frames are then given additional frames to equalize the number of frames across all videos in the dataset.

D. Data Augmentation

Since the total number of videos collected for this dataset is relatively small, consisting of only 900 videos covering 30 sentences, several data augmentation techniques were applied to enhance the variability of our training data. We carefully selected the augmentation techniques based on their compatibility with BISINDO, and made sure that the modifications did not alter the meaning of signs or even violate BISINDO rules. There are two augmentation techniques used in this research, which are:

1) Speed variation: This technique was used to increase the speed of the video by 1.2 times, which is intended to mimic the variability of signing speeds in real-world scenarios. It allows the model to learn and manage different signing speeds.

2) Brightness adjustments: This technique was used to modify the lighting conditions of the videos to simulate different lighting conditions found in different environments, which has been proven to improve model performance on object detection tasks [17]. In this study, brightness adjustments were applied by making the videos 40% darker and 40% brighter than the original video. Examples of brightness variations can be seen in Fig. 1 and Fig. 2. For privacy reasons, faces are covered in the dataset samples shown.

Fig. 1. Original video brightness.

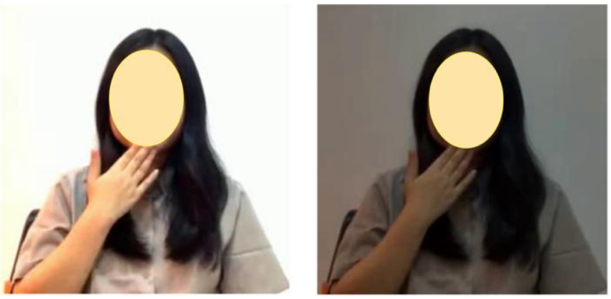

Fig. 2. Video with 40% brighter lighting (left) and video with 40% darker lighting (right).

These techniques were used because they align with the characteristics of the dataset used. Other methods, such as mirroring, were avoided due to BISINDO rules on right and left hand usage, which could alter sign meaning. Other techniques like cropping or adding noise were also not applied, as they might remove essential facial expressions, hand positions, and other details needed for sign recognition.

E. Data Preparation

After all the preprocessing steps have been finished, the dataset was split into three groups, which are: training, validation, and testing sets. The data distribution can be seen in Table II.

TABLE II. ORIGINAL DATA DISTRIBUTION

Data Sets	Percentage	Number of Videos
Training Set	70%	630
Validation Set	15%	135
Testing Set	15%	135
TOTAL	100%	900

As shown in Table II, a 70-15-15 split was applied to the original dataset consisting of 900 videos. Meanwhile, the 2700 augmented videos were included in the training set, as the purpose of augmentation is to enrich the diversity of data available for training, allowing the model to generalize better. The final distribution dataset can be seen in Table III.

TABLE III. FINAL DATA DISTRIBUTION

Data Sets	Number of Videos
Training Set	3330
Validation Set	135
Testing Set	135
TOTAL	3600

F. Model Building and Training

In this phase, we developed a hybrid deep learning model that integrates a 3D CNN and recurrent model (LSTM/BiLSTM) for sentence-level sign language recognition. Since recognizing sign language requires attention and consideration to both manual features (hand shape, position, and movements) and non-manual features (facial expressions, head position, and lip shape) [18], the 3D CNN was utilized to extract spatial and temporal features from the video frames. These extracted features are then passed into two different sequence models, the conventional LSTM and bidirectional LSTM. This model design was chosen to leverage the main strengths of each model, where the 3D CNN serves as the feature extractor, while the LSTM/BiLSTM models capture temporal dependencies within the sign sequence. The model architecture used in this research is presented in Fig. 3, while the configuration of each layer is shown in Table IV. The rows highlighted in blue represent the 3D CNN component of the model, while the rows highlighted in orange represent the LSTM/BiLSTM component. All models were trained from scratch on the BISINDO DKI Jakarta dataset without the use of pre-trained weights or fine-tuning. This was done to ensure that the models are fully adapted to BISINDO sentence-level recognition data.

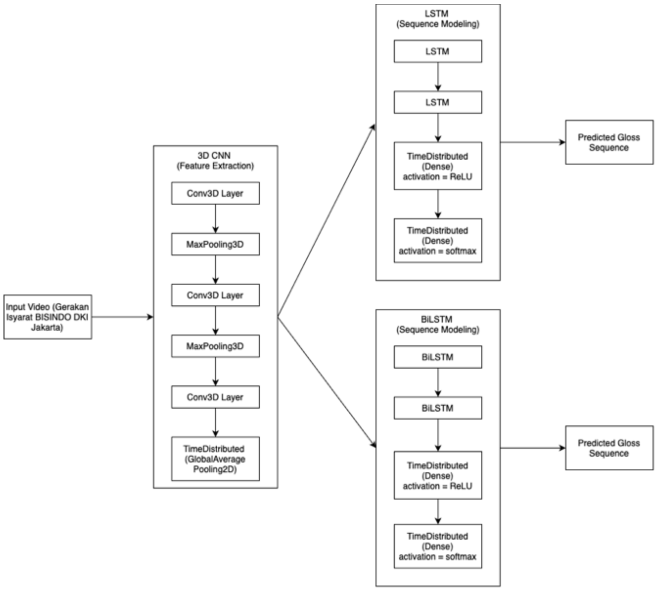
Input Video (Gerakan Isyarat BISINDO DKI Jakarta)
3D CNN
(Feature Extraction)
Conv3D Layer
MaxPooling3D
Conv3D Layer
MaxPooling3D
Conv3D Layer
TimeDistributed (GlobalAverage Pooling2D)
LSTM
(Sequence Modeling)
LSTM
LSTM
TimeDistributed (Dense) activation = ReLU
TimeDistributed (Dense) activation = softmax
Predicted Gloss Sequence
BiLSTM
(Sequence Modeling)
BiLSTM
BiLSTM
TimeDistributed (Dense) activation = ReLU
TimeDistributed (Dense) activation = softmax
Predicted Gloss Sequence

Fig. 3. Model architecture.

TABLE IV. LAYER CONFIGURATION OF THE 3D CNN AND LSTM/BILSTM MODEL

Layer	Configuration	Output Shape
Input 3D CNN	Video frames	(151, 256, 256, 3)
Conv3D	32 filters, kernel (3,3,3), stride (1,1,1), activation ReLU	(151, 256, 256, 3)
MaxPooling3D	Pool size (1, 2, 2)	(151, 128, 128, 32)
Conv3D	64 filters, kernel (3, 3, 3), stride (1, 1, 1), activation ReLU	(151, 128, 128, 64)
MaxPooling3D	Pool size (2, 2, 2)	(75, 64, 64, 64)
Conv3D	128 filters, kernel (3, 3, 3), stride (1, 1, 1), activation ReLU	(75, 64, 64, 128)
TimeDistributed (GlobalAveragePooling2D)		(75, 128)
Input LSTM	3D CNN extracted features	(75, 128)
LSTM/BiLSTM	X units (adjusted based on tuning results)	(75, X)
LSTM/BiLSTM	X units (adjusted based on tuning results)	(75, X)
TimeDistributed Dense (ReLU)	128 units, activation ReLU	(75, 128)
TimeDistributed Dense (Softmax)	45 units, activation Softmax	(75, 45)

As shown in Table IV, firstly the videos containing BISINDO sign language gestures were used as an input for the 3D CNN model. The 3D CNN then performed spatiotemporal feature extraction. The spatial features extracted include hand movements and facial expressions, as these two components are key elements in performing BISINDO, and are essential for identifying sign gestures.

The 3D CNN model used in this study receives video input with dimensions of (151, 256, 256, 3), where:

- 151 represents the number of frames in each video
- 256 x 256 is the resolution of each frame, and
- 3 represents the number of channels (RGB)

This input is processed through several convolutional and pooling layers as shown in Table IV. Then, a TimeDistributed layer with GlobalAveragePooling2D is applied to average the spatial feature values for each video frame individually, preserving the temporal dimension of the video data. This process results in a one-dimensional feature vector for each frame, which can then be passed into the LSTM/BiLSTM model. The output at this stage has the shape (75, 128), where:

- 75 is the number of frames remaining after pooling and downsampling, and
- 128 is the number of features generated for each frame.

To avoid repeating feature extractions due to limited computational resources, the extracted features were stored in a .npy file. These features are then used as an input to the LSTM/BiLSTM model. For sequence modeling, two stacked LSTM/BiLSTM layers were used to capture temporal dependencies. The LSTM processes the frame sequence in one direction, while the BiLSTM processes the sequence in both directions [19]. Stacked LSTMs are used in this study to enhance the model's ability to learn complex relationships in sequential data. In these layers, the output shape becomes (75, X), where 75 is the number of timesteps and X is the number of units in LSTM/BiLSTM which is determined through hyperparameter tuning.

Next, a TimeDistributed Dense layer with 128 units and a ReLU activation function was applied, producing an output of shape (75, 128), where 75 represents the timesteps and 128 represents the number of units in Dense layer. Lastly, a TimeDistributed Dense layer with a Softmax activation function, which is responsible to generate frame-level predictions as probability distributions for every vocabulary and blank token, was applied. It results in an output shape of (75, 45), where 75 represents the timesteps and 45 represents the number of vocabulary in the dataset. During the training, CTC loss function was applied to handle the mismatch between input frames and predicted word lengths. Finally, a decoding process was done using the argmax function to convert the probability outputs into a sequence of predicted words.

After defining the model architecture, hyperparameter tuning was conducted to obtain the most optimal hyperparameter configurations for the 3DCNN-LSTM and 3DCNN-BiLSTM model. This process ensure that the models were trained using the best combination of hyperparameter, so that the models are able to recognize sentences in BISINDO with the lowest possible Word Error Rate (WER). In this study, Bayesian Optimization (BO) was used for hyperparameter tuning due to its efficiency in finding optimal configurations compared to methods like Random Search [20] or Grid Search, while maintaining equal or better model accuracy [21]. The hyperparameters tuned in this research includes the number of LSTM/BiLSTM units, learning rate, batch size, and dropout rate. The predefined ranges for these hyperparameters are shown in Table V.

TABLE V. OPTIMIZED HYPERPARAMETERS

Hyperparameter	Value
LSTM/BiLSTM units	64, 128, 256 units
Learning Rate	0.001, 0.0005, 0.0001
Batch Size	16, 32, 64
Dropout Rate	0.2, 0.3, 0.4

Once the optimal combination of hyperparameter was found for both the 3DCNN-LSTM and 3DCNN-BiLSTM models, the final models were built using these values and training was initiated. This ensured the models were trained under the best configuration, which allows the model to maximize their ability in recognizing word sequences in BISINDO.

G. Evaluation

To evaluate the model against the given input sentences, the metric used in this research is Word Error Rate (WER). This metric is a standard measure commonly used to evaluate the performance of a machine translation system or speech recognition system.

The WER measurement helps evaluate the difference between the output sequence (the identified sentences) and the actual sentences by taking into account substitutions (instances where identified word is different from the actual word intended to be conveyed through sign gestures), deletions (instances where a signed word is missing from the identified output sequence), and insertions (instances where an extra word appears in the output sequence that was not present in the signed gesture). The formula for measuring WER is as follows:

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}$$

where,

- S represents the number of substitutions
- D represents the number of deletions
- I represents the number of insertions
- C represents the number of correctly identified words

By using WER as an evaluation metric, researchers can determine how many errors or mistakes the system makes in identifying sentences in sign language, allowing for a more accurate assessment of the model's performance. A smaller WER value indicates fewer errors in the identified output, meaning that a lower WER reflects better system performance.

IV. RESULTS

This chapter presents the results of the sentence-level SLR model, which was trained and evaluated using the WER metric. Before training the model, hyperparameter tuning was done using Bayesian Optimization to find the best hyperparameter combination for both the 3DCNN-LSTM and 3DCNN-BiLSTM models. The tuning process was carried out with 15 trials for each model. The results of the hyperparameter tuning trials for the LSTM model can be seen in Table VI, and the results for the BiLSTM model can be seen in Table VII.

TABLE VI. LSTM MODEL HYPERPARAMETER TUNING TRIALS

Trial ID	LSTM Units	Dropout	Batch Size	Learning Rate	Score
0	256	0.3	32	0.0005	11.93222332
1	64	0.3	64	0.0005	14.42325211
2	64	0.2	16	0.0005	12.05967999
3	256	0.2	16	0.0005	12.19359112
4	256	0.2	64	0.001	12.75675106
5	128	0.4	16	0.0005	12.39139652
6	128	0.3	16	0.0005	11.87120438
7	256	0.2	64	0.0001	16.38278389
8	128	0.2	64	0.0001	16.72937202
9	128	0.4	64	0.0005	13.32501125
10	256	0.4	64	0.0005	13.03599548
11	128	0.4	64	0.0001	16.78538322
12	256	0.4	32	0.001	12.30996895
13	**128**	**0.2**	**16**	**0.001**	**11.21184921**
14	64	0.3	16	0.001	11.78033161

TABLE VII. BILSTM MODEL HYPERPARAMETER TUNING TRIALS

Trial ID	LSTM Units	Dropout	Batch Size	Learning Rate	Score
0	64	0.3	32	0.0001	12.86169529
1	128	0.4	16	0.0005	9.067760468
2	64	0.4	64	0.0005	11.28538609
3	256	0.4	64	0.001	8.874022484
4	256	0.4	32	0.0001	9.88320446
5	128	0.2	16	0.0005	9.092283249
6	256	0.3	16	0.0005	8.945654869
7	256	0.4	64	0.0005	9.24458313
8	128	0.3	64	0.001	9.18018055
9	256	0.3	64	0.0005	9.229111671
10	256	0.2	64	0.0001	10.96395779
11	128	0.2	32	0.001	9.443668365
12	**256**	**0.4**	**16**	**0.001**	**8.634145737**
13	256	0.2	16	0.001	8.818277359
14	64	0.3	16	0.001	9.163191795

Tables VI and VII show the results of hyperparameter tuning for the LSTM and BiLSTM models, where each trial tests a different combination of LSTM units, dropout rate, batch size, and learning rate. The "Score" column represents the validation loss, meaning that lower values indicate better model performance on unseen data. Based on Tables VI and VII, it can be seen that the best hyperparameter combination for the LSTM model consists of 128 LSTM units, a dropout rate of 0.2, a batch size of 16, and a learning rate of 0.001. Meanwhile, the BiLSTM model achieved its best performance with 256 LSTM units, a dropout rate of 0.4, a batch size of 16, and a learning rate of 0.001. These hyperparameter combinations were then used to build the models for training. Accordingly, the final structure of the 3DCNN-LSTM model can be seen in Fig. 4, and the final structure of the 3DCNN-BiLSTM model can be seen in Fig. 5.

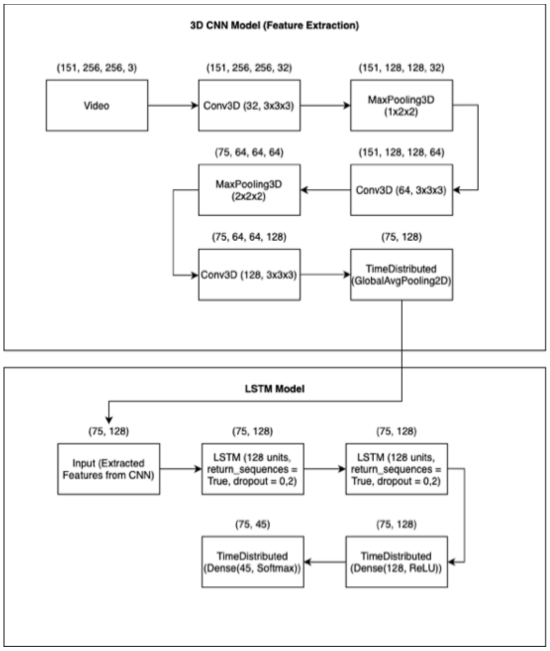
3D CNN Model (Feature Extraction)
(151, 256, 256, 3)
Video
(151, 256, 256, 32)
Conv3D (32, 3x3x3)
(151, 128, 128, 32)
MaxPooling3D (1x2x2)
(151, 128, 128, 64)
Conv3D (64, 3x3x3)
(75, 64, 64, 64)
MaxPooling3D (2x2x2)
(75, 64, 64, 128)
Conv3D (128, 3x3x3)
(75, 128)
TimeDistributed (GlobalAvgPooling2D)
LSTM Model
(75, 128)
Input (Extracted Features from CNN)
(75, 128)
LSTM (128 units, return_sequences = True, dropout = 0,2)
(75, 128)
LSTM (128 units, return_sequences = True, dropout = 0,2)
(75, 128)
TimeDistributed (Dense(128, ReLU))
(75, 45)
TimeDistributed (Dense(45, Softmax))

Fig. 4. Hybrid 3DCNN-LSTM model structure.

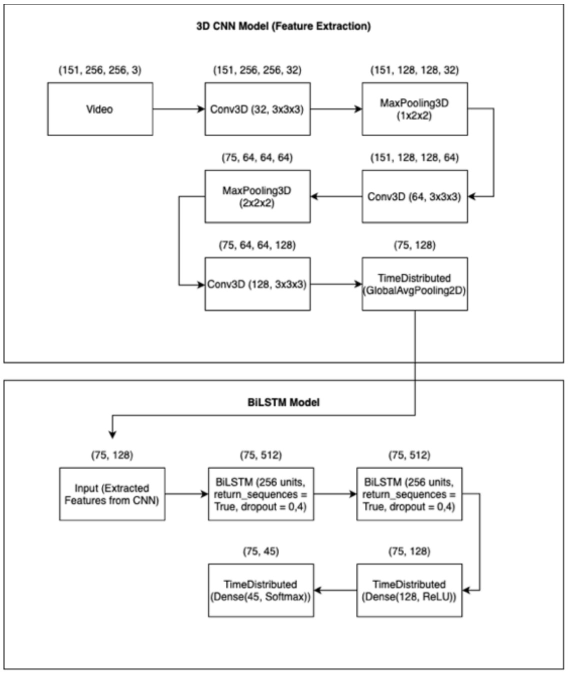
3D CNN Model (Feature Extraction)
(151, 256, 256, 3)
Video
(151, 256, 256, 32)
Conv3D (32, 3x3x3)
(151, 128, 128, 32)
MaxPooling3D (1x2x2)
(151, 128, 128, 64)
Conv3D (64, 3x3x3)
(75, 64, 64, 64)
MaxPooling3D (2x2x2)
(75, 64, 64, 128)
Conv3D (128, 3x3x3)
(75, 128)
TimeDistributed (GlobalAvgPooling2D)
BiLSTM Model
(75, 128)
Input (Extracted Features from CNN)
(75, 512)
BiLSTM (256 units, return_sequences = True, dropout = 0,4)
(75, 512)
BiLSTM (256 units, return_sequences = True, dropout = 0,4)
(75, 128)
TimeDistributed (Dense(128, ReLU))
(75, 45)
TimeDistributed (Dense(45, Softmax))

Fig. 5. Hybrid 3DCNN-BiLSTM model structure.

Each model was trained for 200 epochs. To prevent overfitting, early stopping with a patience of 20 was applied by monitoring the validation loss. The LSTM model achieved its lowest validation loss at epoch 187, with a value of 5.7631. The training and validation loss of the LSTM model can be seen in Fig. 6.

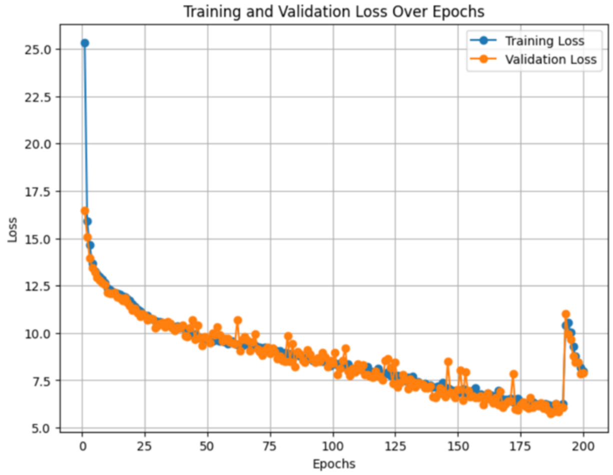

Fig. 6. Training and validation loss of LSTM model.

The graph shows that both training and validation loss decreased steadily, indicating that the model was learning well. However, at epoch 193, there was a sudden spike in both losses, which later dropped again. This spike was likely caused by temporary fluctuations. Since the losses continued to decline afterwards, the model was still learning effectively. In this case, the best weights were used, taken from before the spike occurred.

Meanwhile, the BiLSTM model achieved its lowest validation loss at epoch 183, with a value of 0.2444. The training and validation loss of the BiLSTM model can be seen in Fig. 7.

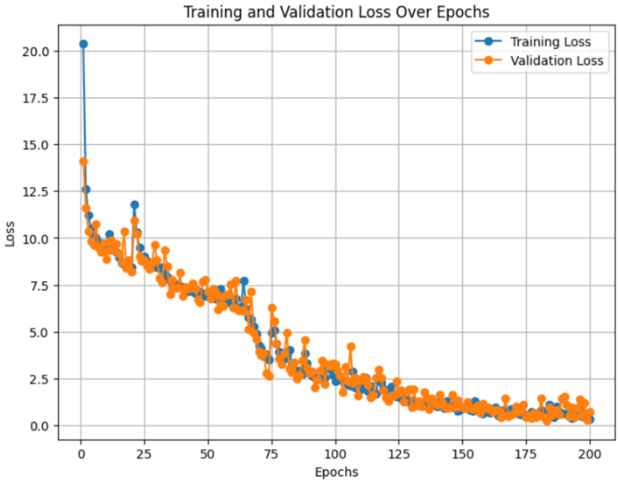

Fig. 7. Training and validation loss of BiLSTM model.

The graph shows a consistent decrease in both training and validation loss, indicating that the model was learning well from the data. Although some fluctuations occurred during training, overall the loss decreased steadily, with the training and validation loss values remaining close throughout the process.

After training and selecting the best weights for both models, we evaluated them on the test set to measure their performance. The predictions from this testing phase were decoded to convert the model's output into sentence form. Once the predicted word sequences were obtained, the model's performance was evaluated using the WER metric, which is calculated based on the number of prediction errors, including substitutions, insertions, and deletions.

In the 3DCNN-LSTM model, the evaluation on the testing set resulted in a relatively high WER of 59.21%. It produced the most errors in substitutions with a total of 211 substitutions, followed by 60 deletions and 28 insertion errors. The error distribution for the LSTM model can be seen in Fig. 8.

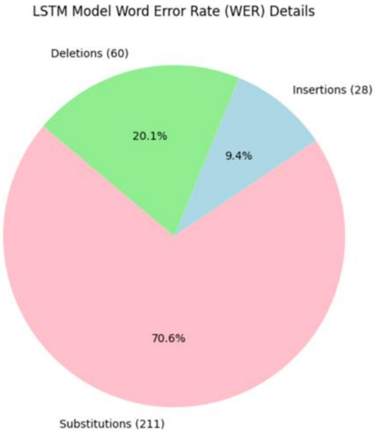

Fig. 8. Error percentage of the LSTM model on the test set.

Meanwhile, the 3DCNN-BiLSTM model achieved a better performance with a significantly lower WER of 2.77%, with the most errors in substitution with a total of 12 substitutions, followed by two insertion errors. The 3DCNN-BiLSTM model did not produce any deletion errors. The error distribution for the BiLSTM model can be seen in Fig. 9.

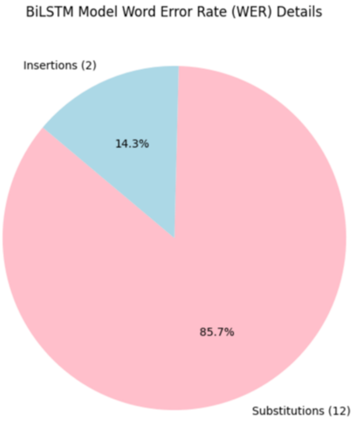

Fig. 9. Error percentage of the BiLSTM model on the test set.

To address the limitation of having only one signer in the dataset, additional signer videos with plain backgrounds and clear visuals were collected from publicly available sources on the internet to serve as a supplementary test set. We successfully collected seven videos of BISINDO DKI Jakarta sentences to be used for the additional test.

In testing using the additional signer videos collected from the internet, the LSTM model achieved a WER of 120% with 11 substitution errors, followed by 5 insertions and 2 deletions errors. Meanwhile, the BiLSTM model achieved a WER of 60%, with 9 substitution errors without insertion and deletion errors.

V. Discussion

Both 3D CNN-LSTM and 3D CNN-BiLSTM models were trained using the hyperparameters determined from tuning, ensuring that each model was optimized for its architecture. Both models were trained for the same number of epochs, which is 200 epochs, to allow a fair comparison. The average training time per epoch was 8.66 seconds for the LSTM and 11 seconds for BiLSTM model. The BiLSTM model contains a larger number of trainable parameters than the LSTM, which contributes to the slightly longer training time. While BiLSTM requires slightly more time due to its bidirectional architecture, the difference is small and does not significantly affect the fairness of the comparison.

Based on the evaluation results of the LSTM and BiLSTM models, there is a significant difference in WER between the two models. The 3D CNN-BiLSTM model shows a drastically lower WER compared to the 3D CNN-LSTM model. As seen from the training loss, the LSTM loss was still decreasing at the end of 200 epochs, indicating that it could potentially achieve slightly better performance if trained for more epochs. However, the BiLSTM still outperforms the LSTM due to its ability to capture context in both directions.

In sentence-level SLR, the meaning of a gesture often depends on both what comes before and after, creating context across the whole sentence. While the LSTM can only utilize past frames, the BiLSTM may look at the sequence in both directions. This bidirectional processing helps BiLSTM to distinguish gestures that might look similar on their own and better capture subtle differences in hand movements and facial expressions. Even if the LSTM is trained longer, LSTM is still limited in using future context, which explains the drastic performance difference between the models. This also shows that bidirectional modeling is so important for sentence-level SLR.

However, the significantly low WER on the BiLSTM model may also be due to the lack of signer variation in the dataset, meaning the model has effectively learned the patterns of a single individual, which allows it to predict word sequences in the test set very well. This becomes clear when testing on videos from new signers, where the WER is higher than on the original test set. However, this should not be considered as the main benchmark, as uncontrolled factors in videos collected from social media can affect the model's ability to recognize sign language. The differences in signing style may also contribute to the model's difficulty in accurately recognizing gestures in videos from new signers. Although the model cannot yet fully recognize the word sequences in sign language sentences, it can be seen that it can fairly well recognize and identify sentence length. This is demonstrated by the small number of insertion and deletion errors made by both models, while both models make more substitution errors. This demonstrates that overall sentence length can be identified well, although errors in sign recognition still occur, which can be due to various factors, such as similar gestures.

Nevertheless, compared to previous similar work, such as sentence-level SLR using 3D CNN-BiRNN model, the 3D CNN-BiLSTM model still shows a lower WER, even when testing on videos from different signers. This improvement is likely due to the LSTM's gating mechanisms, which better handle vanishing gradients, allowing the model to remember important information over longer sequences better than BiRNN.

VI. Conclusion

This study presents a novel approach to sentence-level BISINDO DKI Jakarta sign language recognition by combining a 3D CNN model for spatial and temporal feature extraction, followed by a LSTM/BiLSTM network to capture sequential dependencies, enhanced by CTC, which allows for sentence-level sign language recognition without the need for word-level annotations. Both 3DCNN-LSTM and 3DCNN-BiLSTM model was trained and evaluated on the 30 BISINDO sentence videos consisting of everyday phrases, with data augmentation applied to improve generalization and hyperparameters tuned to optimize performance.

The research was conducted through several stages. First, the feature extraction was performed by the 3D CNN model, with the resulting features stored in a .npy file. These features are then passed to two different models – LSTM and BiLSTM – for training and testing in sentence-level BISINDO recognition. Both models were built using optimal hyperparameter configurations obtained through Bayesian Optimization. From the experiments, several key findings can be drawn:

1) The 3DCNN-LSTM model was not able to achieve a good recognition performance, indicated by a relatively high WER of 59.21% on the testing set. When evaluated on new signer videos, the model produced an even higher WER of 120%, demonstrating that it was unable to effectively recognize sign language, particularly when faced with data from different signers.

2) The 3DCNN-BiLSTM model showed significantly better recognition performance than the 3DCNN-LSTM. It achieved a low WER of 2.77% on the testing set, indicating strong performance for recognizing sentences from a single signer. However, when tested on new signer videos, it achieved a higher WER of 60%. While it is still not highly accurate in recognizing unseen signers, the BiLSTM outperformed the LSTM. This is likely due to BiLSTM's ability to capture contextual information in both forward and backward directions.

3) The combination of 3D CNN as a feature extractor and BiLSTM as a sequence model demonstrated promising results for sentence-level BISINDO recognition in the testing set. However, this outstanding performance is still influenced by the limitations of the dataset, especially the limited number of signers, which causes the model to overfit to specific movement patterns.

4) Although neither the 3DCNN-LSTM nor the 3DCNN-BiLSTM achieved perfect recognition of BISINDO sentences, both models were relatively effective in capturing sentence length. This was reflected by fewer insertion and deletion errors compared to substitution errors.

One of the main limitations in this study is the very limited dataset size, which only involves one non-native signer, so the model's ability to generalize to various variations of signer gestures when demonstrating sign language cannot be fully proven. Therefore, in the future, it is recommended for researchers to develop the dataset by adding more signers so that the model can learn the variations of gestures of several signers who have their own unique characteristics. Furthermore, the number of sentence samples should be increased to allow the model to learn from a broader variety of expressions. By expanding the dataset, the resulting model will be able to recognize a wider range of sign language gestures.

Furthermore, based on the excellent results of the 3DCNN-BiLSTM model evaluation in recognizing sign language sentences on a single signer test set, the development of personalized sign language recognition can also be considered in future research. As the number of datasets and the variety of signers increase, researchers can also develop this recognition model into a system with a broader scope for public use. For example, this research can be further developed into a web-based or mobile application that can be used as a medium for learning sign language in the community, to become a reliable communication medium for the deaf and hearing communities.

Acknowledgment

The authors would like to thank Bina Nusantara University, Jakarta, for their support in providing the research facilities.

Authors' Contribution

Katriel Larissa Wiguna: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Resources, Data Curation, Writing – Original Draft, Visualization. Rojali: Writing – Review & Editing, Supervision.

Data Availability

Data supporting this study are not publicly available because they contain identifiable facial information, which raises privacy concerns.

References

[1] A. S. Nugraheni, A. P. Husain, and H. Unayah, "Optimalisasi Penggunaan Bahasa Isyarat Dengan SIBI DAN BISINDO Pada Mahasiswa Difabel Tunarungu Di Prodi PGMI UIN Sunan Kalijaga," Jurnal Holistika, vol. 5, no. 1, p. 28, Feb. 2023, doi: 10.24853/holistika.5.1.28-33.

[2] A. Wadhawan and P. Kumar, "Sign Language Recognition Systems: A Decade Systematic Literature Review," Archives of Computational Methods in Engineering, vol. 28, no. 3, pp. 785–813, May 2021, doi: 10.1007/s11831-019-09384-2.

[3] R. S. L. Murali, L. D. Ramayya, and V. A. Santosh, "Sign Language Recognition System Using Convolutional Neural Network And Computer Vision," International Journal of Engineering Innovations in Advanced Technology, vol. 4, no. 4, pp. 137–142, Dec. 2022.

[4] M. C. Ariesta, F. Wiryana, Suharjito, and A. Zahra, "Sentence Level Indonesian Sign Language Recognition Using 3D Convolutional Neural Network and Bidirectional Recurrent Neural Network," in 2018 Indonesian Association for Pattern Recognition International

Conference (INAPR), IEEE, Sep. 2018, pp. 16–22. doi: 10.1109/INAPR.2018.8627016.

[5] J. Shin et al., "Korean Sign Language Recognition Using Transformer-Based Deep Neural Network," Applied Sciences, vol. 13, no. 5, p. 3029, Feb. 2023, doi: 10.3390/app13053029.

[6] P. Vyavahare, S. Dhawale, P. Takale, V. Koli, B. Kanawade, and S. Khonde, "Detection and Interpretation of Indian Sign Language Using LSTM Networks," Journal of Intelligent Systems and Control, vol. 2, no. 3, pp. 132–142, Jul. 2023, doi: 10.56578/jisc020302.

[7] G. Latif, N. Mohammad, R. AlKhalaf, R. AlKhalaf, J. Alghazo, and M. Khan, "An Automatic Arabic Sign Language Recognition System based on Deep CNN: An Assistive System for the Deaf and Hard of Hearing," International Journal of Computing and Digital Systems, vol. 9, no. 4, pp. 715–724, Jul. 2020, doi: 10.12785/ijcds/090418.

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

[9] G. S. Sajja, S. R. Addula, M. K. Meesala, and P. Ravipati, "Optimizing inventory management through AI-driven demand forecasting for improved supply chain responsiveness and accuracy," 2025, p. 050003. doi: 10.1063/5.0275697.

[10] Badan Pusat Statistik, "Kepadatan Penduduk menurut Provinsi (jiwa/km2), 2021."

[11] A. Aljabar and S. Suharjito, "BISINDO (Bahasa Isyarat Indonesia) Sign Language Recognition Using CNN and LSTM," Advances in Science, Technology and Engineering Systems Journal, vol. 5, no. 5, pp. 282–287, 2020, doi: 10.25046/aj050535.

[12] R. A. Alawwad, O. Bchir, and M. Maher, "Arabic Sign Language Recognition using Faster R-CNN," International Journal of Advanced Computer Science and Applications, vol. 12, no. 3, 2021, doi: 10.14569/IJACSA.2021.0120380.

[13] L. Arisandi and B. Satya, "Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network," Jurnal Sistem Cerdas, vol. 5, no. 3, pp. 135–146, Dec. 2022, doi: 10.37396/jsc.v5i3.262.

[14] D. Kothadiya, C. Bhatt, K. Sapariya, K. Patel, A.-B. Gil-González, and J. M. Corchado, "Deepsign: Sign Language Detection and Recognition Using Deep Learning," Electronics (Basel), vol. 11, no. 11, p. 1780, Jun. 2022, doi: 10.3390/electronics11111780.

[15] T. H. Noor et al., "Real-Time Arabic Sign Language Recognition Using a Hybrid Deep Learning Model," Sensors, vol. 24, no. 11, p. 3683, Jun. 2024, doi: 10.3390/s24113683.

[16] E. N. Siagian, "Kata Berfrekuensi Tinggi dalam Pembelajaran BIPA Pemula," Ranah: Jurnal Kajian Bahasa, vol. 9, no. 2, p. 188, Dec. 2020, doi: 10.26499/rnh.v9i2.2320.

[17] A. M. Abdulghani, M. M. Abdulghani, W. L. Walters, and K. H. Abed, "Data Augmentation Using Brightness and Darkness to Enhance the Performance of YOLO7 Object Detection Algorithm," in 2023 Congress in Computer Science, Computer Engineering, &amp; Applied Computing (CSCE), IEEE, Jul. 2023, pp. 351–356. doi: 10.1109/CSCE60160.2023.00061.

[18] N. B. Ibrahim, H. H. Zayed, and M. M. Selim, "Advances, Challenges and Opportunities in Continuous Sign Language Recognition," Journal of Engineering and Applied Sciences, vol. 15, no. 5, pp. 1205–1227, Dec. 2019, doi: 10.36478/jeasci.2020.1205.1227.

[19] R. L. Abduljabbar, H. Dia, and P.-W. Tsai, "Unidirectional and Bidirectional LSTM Models for Short-Term Traffic Prediction," J Adv Transp, vol. 2021, pp. 1–16, Mar. 2021, doi: 10.1155/2021/5589075.

[20] R. Turner et al., "Bayesian Optimization is Superior to Random Search for Machine Learning Hyperparameter Tuning: Analysis of the Black-Box Optimization Challenge 2020," Apr. 2021.

[21] A. Stuke, P. Rinke, and M. Todorović, "Efficient hyperparameter tuning for kernel ridge regression with Bayesian optimization," Apr. 2020.