Jurnal Aplikasi Sains Data
Journal homepage: https://jasid.upnjatim.ac.id/index.php/jasid

Application of Convolutional Neural Network (CNN) for Web-Based Translation of Indonesian Text into Sign Language

Diajeng Sekar Prameswari[1], Larasati[2], Muhammad Naswan Izzudin Akmal[3], Prismahardi Aji Riyantoko[4], Dwi Arman Prasetya[5,*]

[1,2,3,5]Faculty of Computer Science, UPN "Veteran" Jawa Timur

[4]Faculty of Computer Science Okayama University Japan

[3]pnai2m3s@s.okayama-u.ac.jp, [5,*]arman.prasetya.sada@upnjatim.ac.id

ABSTRACT

Abstract— Communication for the deaf and hard of hearing is often hindered by the limited number of sign language interpreters. This research aims to develop a web-based text-to-text sign language translation system using Convolutional Neural Networks (CNN) to bridge this communication gap. The system is built with the ASL Alphabet dataset containing 87,000 images from 29 classes (A-Z, SPACE, DELETE, NOTHING). The CNN model was designed with three convolutional layers and trained for 15 epochs using 80% of the data, while 20% of the data was used for testing. The user interface was developed using Streamlit for ease of use. Training results showed a training accuracy of 98.96% and a validation accuracy of 98.61% at the 15th epoch. Model evaluation yielded an overall accuracy of 98%, with high precision, recall, and F1-score values for most classes. This research demonstrates the significant potential of CNN in developing automatic sign language translators, which is expected to improve information accessibility and inclusivity for the deaf community.

Keywords: Sign language, *Convolutional Neural Network* (CNN), *ASL Alphabet*, Streamlit, *Machine Learning*

I. INTRODUCTION

In the daily lives of deaf and hard-of-hearing persons, sign language functions as the dominant form of communication. This language has its own structure and rules that differ from spoken language, and it plays a crucial role in bridging communication with the general public [1]. Globally, more than 300 sign languages are used by approximately 72 million deaf individuals, and 80% of them live in developing countries [2]. As of May 2025, only 81 out of 195 countries have officially recognized their national sign language, including Indonesia since 2016 [3]. Domestically, based on the 2022 Population Census, there are 159,918 individuals, or 17.12% of the total disabled population, who experience sensory limitations, including the deaf and hard-of-hearing [4]. However, the limited number of sign language interpreters acts as a barrier to ensuring equal access to information and services, especially in the education, public services, and digital media sectors.

In recent years, various efforts have been made to bridge the communication gap between deaf individuals and the wider community. One such effort is through the use of artificial intelligence (AI)-based technology to develop automatic sign language translation systems. This system is expected to

* *Corresponding author.*
E-mail address: arman.prasetya.sada@upnjatim.ac.id
10.33005/jasid.v1i2.12

increase information accessibility while helping the public understand and learn sign language [5]. Along with the development of computer technology, deep learning algorithms like Convolutional Neural Networks (CNN) are widely used to recognize visual patterns, including hand movements in sign language. One form of its application is the development of applications or systems capable of recognizing hand gestures in real-time and translating them into spoken language or text [6].

The effectiveness of the CNN algorithm in developing sign language translator systems has been proven through a number of previous studies. Nurhayati et al. [7] achieved an accuracy of 97.2% in a real-time hand gesture recognition system using CNN. Another study by Alfikri et al. [8] developed an Android application with training accuracy up to 97% over 200 epochs, although the accuracy decreased to 73% after deployment. Meanwhile, Pramono et al. [9] applied CNN for Indonesian sign language translation with an average detection accuracy of 70.2%. These findings indicate that CNN has high potential in sign language recognition, although the final system performance is highly influenced by implementation conditions and training parameters.

Based on the description above, this research is titled "Development of a Web-Based Text-to-Sign Language Translation System Using Convolutional Neural Network (CNN)". The system is built to automatically convert Indonesian language sentence input into a sequence of sign language images with a Streamlit-based interface. CNN is used to process and recognize visual patterns from sign language images, while Streamlit is used as the user interface platform. With this system, it is hoped that the communication process between deaf individuals and the wider community can become more inclusive and efficient.

II. RESEARCH METHODS

A. Data

This research uses the ASL Alphabet dataset, which is a collection of image data of alphabet letters in American Sign Language. The training dataset consists of 87,000 images with a resolution of 200x200 pixels, divided into 29 classes: 26 classes for letters A-Z and 3 additional classes: SPACE, DELETE, and NOTHING. These three additional classes are very helpful in real-time applications and sign language classification. The total overall data is 87,000 images. Sample data examples are presented in Fig. 1.

Figure 1. Sample Dataset

B. Research Stages

This research is an experimental study that uses the Convolutional Neural Network (CNN) method to develop a web-based Indonesian text-to-sign language translation system. The research stages are

shown in Fig. 2.

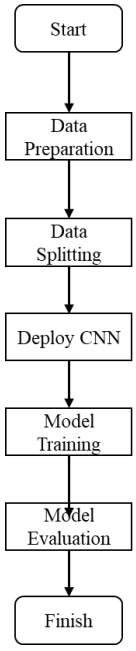

Figure 2. Flowchart of Research

In the data and label preparation stage, the ASL Alphabet dataset was processed by creating a function to display images of letters A to Z and assign labels for each letter class. This process included reading images from their respective class folders, resizing images to 128x128 pixels, converting them to RGB format, and storing image data and labels separately.

Next, the data was divided into two subsets: 80% for training data (x_train, y_train) and 20% for testing data (x_test, y_test), using the train_test_split method with shuffle=True and random_state=42 parameters to maintain consistency. After division, the data sizes became: x_train (69600, 128, 128, 3), y_train (69600,), x_test (17400, 128, 128, 3), and y_test (17400,).

The architectural foundation of this research is a CNN, a powerful neural network type that leverages convolution filters to learn input features. CNNs are particularly adept at processing image data and generating predictions for predefined classes, a capability previously validated in sign language prediction with accurate results [7].

For this study, the CNN was meticulously crafted using the Keras framework [9]. Its design incorporates three Conv2D layers, utilizing both 32 and 64 filters with kernel sizes of (5,5) and (3,3) respectively. Each of these is succeeded by a MaxPooling2D layer, serving to extract key features and reduce data dimensionality. Subsequently, a Flatten layer transforms the output before it enters a dense layer of 128 neurons, activated by ReLU. The final output layer, equipped with 29 neurons and softmax activation, handles multi-class classification (as depicted in Fig. 3). The model was configured for training using the Adam optimizer [11][12], the cross-entropy loss function, and the accuracy metric.

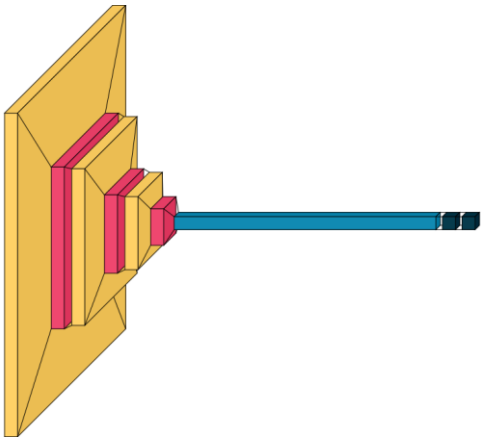

Figure 3. Visualization of CNN Model

Model development involved training over 15 epochs with 32-sample batches on designated training data, simultaneously validating against testing data to track accuracy and loss evolution. Post-training, performance assessment comprised visualizing accuracy and loss trends for both training and validation sets, alongside a comprehensive classification report detailing per-class precision, recall, f1-score, and support to quantify classification efficacy.

III. RESULT AND DISCUSSIONS

A. Model Architecture

The CNN model developed uses three convolutional layers (Conv2D) and three pooling layers (MaxPooling2D), followed by a flatten layer and two dense layers, with a total of 1,667,357 parameters. The final layer consists of 29 neurons representing the 26 letters of the ASL alphabet, as well as three additional classes for some common symbols. Table 1 shows the model architecture.

TABLE I. MODEL ARCHITECTURE

Layer	Output Shape	Parameter count
Conv2D (32 filter)	(124, 124, 32)	2,432
MaxPooling2D	(62, 62, 32)	0
Conv2D (64 filter)	(60, 60, 64)	18,496
MaxPooling2D	(30, 30, 64)	0
Conv2D (64 filter)	(28, 28, 64)	36,928
MaxPooling2D	(14, 14, 64)	0
Flatten	(12544)	0
Dense (128 neuron)	(128)	1,605,760
Output Dense (29)	(29)	3,741

B. Model Training Result

The model was trained for 15 epochs with a batch size of 32. The accuracy and loss graphs show a significant improvement in model performance at the beginning of training and stabilization towards the end. Training accuracy increased from 52.01% in the first epoch to 98.96% in the 15th epoch. Meanwhile, validation accuracy peaked at 98.61%, indicating that the model did not experience significant overfitting.

TABLE II. TRAINING RESULT

Epoch	Train Accuracy	Validation Accuracy	Validation Loss
1	52.01%	91.83%	0.2547
5	97.70%	98.16%	0.0601
10	98.30%	98.54%	0.0787
15	**98.96%**	**97.84%**	**0.1064**

C. Model Evaluation

In Fig. 6, it is observed that the model training process shows a consistent increase in accuracy and a parallel decrease in loss values for both training and validation data as epochs increase. Accuracy reaches approximately 98% on both datasets, indicating that the model has learned well and is not overfitting. This balanced decrease in loss reinforces the model's ability to generalize patterns on new, unseen data. Thus, the training process is effective and the model demonstrates adequate performance stability.

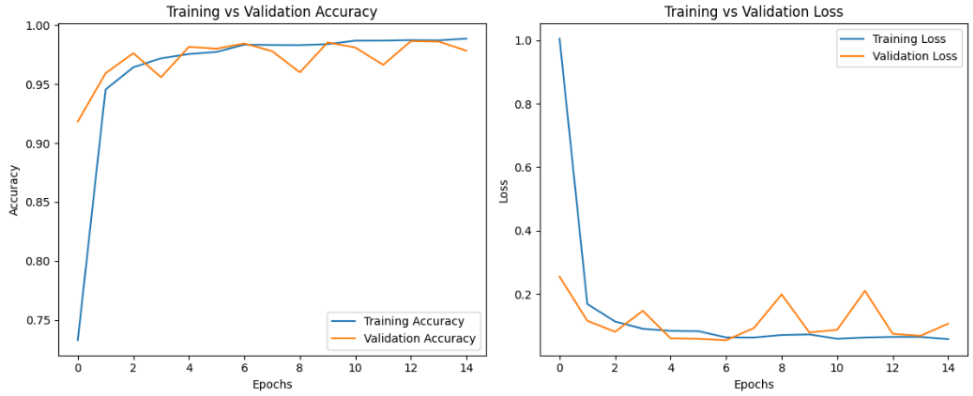

Figure 6. Training Loss Graph

The model evaluation results based on the classification report in Table 3 show high precision, recall, and f1-score values for almost all classes, with the majority above 0.95. Classes J, Q, and F have the best performance with f1-scores reaching 0.99 to 1.00, while class M has the lowest f1-score of 0.95, which is still considered very good. The overall model accuracy reached 98%, supported by balanced macro average and weighted average values of 0.98. This indicates that the model has consistent performance and is not biased towards specific classes, making it very reliable for multi-class classification on large datasets.

TABLE III. MODEL EVALUATION

Label	Precision	Recall	F1-Score
A	0.98	0.99	0.98
B	0.99	0.95	0.97
C	1.00	0.97	0.98
D	0.95	0.99	0.97
E	0.95	0.98	0.97
F	0.99	0.99	0.99
G	0.99	0.98	0.99

H	0.99	0.99	0.99
I	0.98	0.98	0.98
J	0.99	1.00	0.99
K	0.99	0.97	0.98
L	0.98	0.99	0.99
M	0.99	0.90	0.95
N	0.94	1.00	0.97
O	0.98	0.97	0.98
P	0.99	0.99	0.99
Q	0.98	1.00	0.99
R	0.99	0.97	0.98
S	0.95	0.97	0.96
T	0.98	0.96	0.97
U	0.95	0.99	0.97
…	…	…	…
Accuracy	0.98	0.99	0.98
Macro avg			
Weighted avg	0.99	0.95	0.97

D. Confusion Matrix

To learn more about the model's performance, an evaluation was conducted using a confusion matrix, as shown in Fig. 8 This confusion matrix shows the model's correct and incorrect prediction results when predicting sign language with letters A-Z and three other symbols. It can be seen that the model can predict all classes with good accuracy.

Some classes show excellent classification results. For example, the letters N, V, and W can be predicted well and obtain 600 correct predictions. Three additional tokens such as SPACE, DELETE, and NOTHING can also be predicted well; these three tokens are very helpful in the use of sign language, as they can indicate the end of a sentence. Nevertheless, there are still classification errors in some letters. For example, the letter M is often predicted as the letter N, and the letter U is classified as the letter V. These errors generally occur because the sign language shapes of these letters are very similar. This issue could be a subject for further study, for example, by adding data augmentation techniques to improve model accuracy.

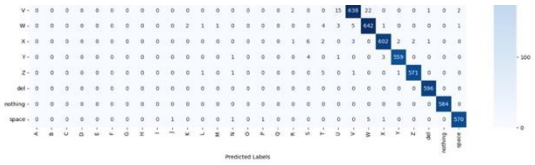

Figure 8. Model Confusion Matrix

E. Web Application

This web application was developed using the Streamlit framework, chosen for its ability to simplify the development process of Python-based web applications. Streamlit allows for rapid integration between the user interface and machine learning model-based data processing, making it highly suitable for the needs of this application.

The application is designed to provide two main functions. The first function is the translation of Indonesian text into a visual representation in the form of sign language. This process is carried out by converting each letter of the word or sentence entered by the user into images of letters in sign language. Fig. 9 shows the interface display of this feature, where the sentence "Indonesia Jaya" has been successfully translated into a series of sign language images corresponding to its constituent letters.

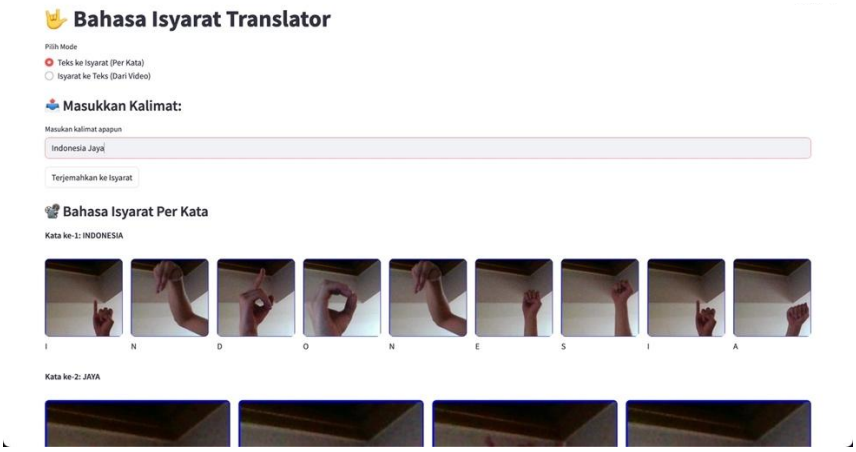

Figure 9. Screen Capture of Text-to-Sign in Streamlit

The second function is the translation of sign language video into Indonesian text. Users can upload videos in MP4 or AVI format. After the video is uploaded, the system will process it gradually by dividing the video into image frames. Each frame is then analyzed and classified using a pre-trained CNN model to recognize hand gestures. The classification results from each frame will be arranged into a series of texts, so that the meaning of the sign language can be understood in written form. For example, in Fig. 10, a sign language video with the words "SELAMAT PAGI" is used. The model successfully detected the gesture and provided the corresponding translation.

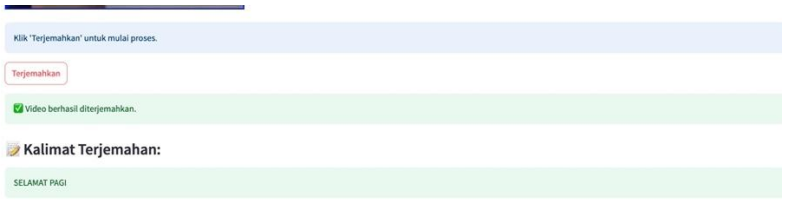

Figure 10. Screen Capture of Isyarat-to-text in Streamlit

The sign language video translation process takes 5 seconds for a 2-second video with a sentence length of 13 letters. One limitation in web application implementation is that it is limited to videos where only the hands have been cropped.

CONCLUSION

This research successfully designed and assessed a web-based tool for converting text to American Sign Language (ASL) images and ASL videos to text. Employing a Convolutional Neural Network (CNN) trained on the ASL Alphabet dataset, the system demonstrated high performance with a validation accuracy of 98.61% and a test accuracy of 98%. Other metrics, including precision, recall, and F1-score, consistently surpassed 0.95, indicating the model's effectiveness. Developed using Streamlit, the application primarily facilitates letter-by-letter translation from Indonesian text to ASL and vice versa. A key limitation of the current system is its restriction to individual letter translation; future work could explore models like Long Short-Term Memory (LSTM) or transformer networks to interpret sign language gestures at the word or sentence level.

REFERENCES

[1] R. Brindha, P. Renukadevi, D. Vathana, and J. D, “Sign Language Interpreter,” in 2022 International Conference on Inventive Computation Technologies (ICICT), IEEE, Jul. 2022, pp. 1024–1028. doi: 10.1109/ICICT54344.2022.9850486.

[2] National Geographic Society, “Sign Language,” National Geographic Society. Accessed: Jun. 05, 2025. [Online]. Available: https://education.nationalgeographic.org/resource/sign-language/?utm_source=chatgpt.com

[3] World Federation of the Deaf (WFD), “The Legal Recognition of National Sign Languages,” World Federation of the Deaf. Accessed: Jun. 05, 2025. [Online]. Available: https://wfdeaf.org/the-legal-recognition-of-national-sign-languages/?utm_source=chatgpt.com&_s=indonesia

[4] R. Maheswara, “Jumlah Penyandang Disabilitas di Indonesia 2022 Berdasarkan Jenis Keterbatasan,” Humaniora. Accessed: Jun. 05, 2025. [Online]. Available: https://dataloka.id/humaniora/1967/jumlah-penyandang-disabilitas-di-indonesia-2022-berdasarkan-jenis-keterbatasan/?utm_source=chatgpt.com#google_vignette

[5] D. Ponnappa and B. G. Jairam, “A Comparative Study on Sign Language Translation Using Artificial Intelligence Techniques,” 2023, pp. 359–368. doi: 10.1007/978-981-19-8742-7_30.

[6] S. Meshram, R. Singh, P. Pal, and S. K. Singh, “Convolution Neural Network based Hand Gesture Recognition System,” in 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT), IEEE, Jan. 2023, pp. 1–5. doi: 10.1109/ICAECT57570.2023.10118267.

[7] O. Dwi Nurhayati, D. Eridani, and M. Hafiz Tsalavin, “SISTEM ISYARAT BAHASA INDONESIA (SIBI) METODE CONVOLUTIONAL NEURAL NETWORK SEQUENTIAL SECARA REAL TIME A REAL-TIME INDONESIAN LANGUAGE SIGN SYSTEM USING THE CONVOLUTION NEURAL NETWORK METHOD,” Jurnal teknologi Informasi dan Ilmu Komputer (JTIIK)), vol. 9, pp. 819–828, 2022, doi: 10.25126/jtiik.202294787.

[8] R. Haris Alfikri et al., “PEMBANGUNAN APLIKASI PENERJEMAH BAHASA ISYARAT DENGAN METODE CNN BERBASIS ANDROID,” 2022. [Online]. Available: https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

[9] I. Muslim Pramono, imatun Niswati, and A. Agustina, “MODEL PENERJEMAH BAHASA ISYARAT DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN),” jakarta, Jan. 2024.

[10] F. Chollet and Others, ‘Keras’, 2015. [Online]. Available: https://keras.io.

[11] D. P. Kingma and J. Ba, ‘Adam: A Method for Stochastic Optimization’, arXiv [cs.LG]. 2017.

[12] A. Muhaimin, D. D. Prastyo and H. Horng-Shing Lu, "Forecasting with Recurrent Neural Network in Intermittent Demand Data," *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2021, pp. 802-809, doi: 10.1109/Confluence51648.2021.9376880.