Sign Language Translator for SIBI

Ezra Arya Wijaya
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
ezra.wijaya001@binus.ac.id

Samuel Benediktus Meliala
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
samuel.meliala@binus.ac.id

Muhammad Fadlan Hidayat
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
muhammad.hidayat@binus.edu

Irene Anindaputri Iswanto
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
irene.iswanto@binus.edu

***Abstract*—Deaf people face communication barriers and sign language is essential for them, especially in Indonesia. Indonesia has two main sign languages: BISINDO and SIBI. BISINDO is community developed, while SIBI is adapted from American Sign Language for standardization. In this study, we develop a gesture recognition system to translate SIBI using a convolutional neural network (CNN). Utilizing MobileNetV2, our model achieved over 99.7% accuracy in controlled tests and over 99.9% accuracy on mobile devices. However, live camera tests revealed challenges with accuracy across different scripts. Further optimization is needed. This work advances SIBI sign language recognition and enhances communication tools for the deaf community in Indonesia.**

Keywords—SIBI, BISINDO, Sign Language Traanslation, Gesture Recognition, Convolutional Neural Networks, MobileNetV2

I. INTRODUCTION

Deafness is a condition in which individuals have very low or no hearing. It can be caused by a variety of factors, such as genetics, ear infections, or trauma to the auditory system. Deaf people can experience barriers to communication and social interaction, especially with individuals who have normal hearing.

Sign language serves as the main communication tool for the deaf to convey ideas, thoughts, and feelings. In Indonesia, the development of sign language has had its ups and downs. At first, sign language was not recognised as an official language and its use was limited to the deaf community. However, as awareness and understanding of deafness increased, sign language began to gain recognition and be used more widely.

Indonesia has a great population if the deaf, estimated to be about 2.1 million individuals. Despite the vast number, only a portion of these has access proficiency in sign language. There are two known primary sign language:

1. Indonesian Sign Language (BISINDO): A natural sign language that developed organically in the Indonesian deaf community. BISINDO has dialectal variations in different regions, reflecting the richness of Indonesian culture.
2. Sistem Isyarat Bahasa Indonesia (SIBI): A sign language adapted from American Sign Language (ASL). SIBI was created with the aim of unifying sign languages across Indonesia and facilitating communication between deaf people from different regions.

SIBI and BISINDO have some similarities, such as grammatical structures and basic vocabulary. However, there are some key differences, namely:

1. Origin: SIBI was adapted from ASL, while BISINDO developed naturally in the Indonesian deaf community.
2. Usage: SIBI is generally used in special schools (SLB) and in formal communication, while BISINDO is used more in everyday communication in the deaf community.
3. Variation: SIBI has fewer variations than BISINDO, as it is designed to standardise sign language across Indonesia.

The emergence of Artificial Intelligence (AI) brought a breath of fresh air for the deaf with the presence of Gesture Recognition that allows sign language to be translated real-time. Gesture Recognition is a computing process that attempts to recognize and interpret human gestures through a set of algorithms. Mentioned in [1], Gesture Recognition has been applied on various of field including sign language translation, robot control, medical system, etc. In Indonesia, Gesture Recognition technology for sign language translation has primarily focused on BISINDO (Bahasa Isyarat Indonesia). This research aims to bridge the gap by developing gesture recognition that can also translate SIBI (Sistem Isyarat Bahasa Indonesia).

Although gesture recognition technology has advanced, one of the major challenges is the lack of applications that can translate SIBI in real-time. As such, this research is projected towards the development of an application using Convolutional Neural Networks to translate SIBI in advancing the recognition of SIBI sign language and further increasing the tools for communication among Indonesia's deaf community.

II. LITERATURE REVIEW

A. Previous work in gesture recognition for sign language translation

Gesture Recognition identifies the hand gesture to recognize the sign language and translates them. There have been several ways to implement gesture recognition for this field. In the paper "Gesture recognition for Indonesian Sign Language (BISINDO)" [4] conducted research to implement Hidden-Markov Model (HMM) as the neural network for Gesture Recognition. The experiment concluded that implementing HMM to gesture recognition results in

accuracy of 60\%-70\%. The mediocre accuracy is the result of inaccuracy from the data of the performed signed language.

Another paper titled "Indonesian Sign Language Recognition using Convolutional Neural Network" [5] conducted research on deep learning approach using Convolutional Neural Network (CNN) to recognize BISINDO. The model when tested using test data achieved an accuracy of 98.3\%, precision of 98.3\%. This research conducted gives an excellent result, one of the takeaways than can be taken from this experiment is lighting condition and perspective can impact the result.

Another paper titled "Translating SIBI (Sign System for Indonesian Gesture) Gesture-to-Text in Real-Time using a Mobile Device" [3] conducted research on translating SIBI (Sign System for Indonesian Gesture) in real-time using MobileNetV2, CRF, and LSTM loaded into one interface. The result showed an excellent accuracy of 90.56\% with the average translation time of 20 seconds.

B. Gesture Detection

Gesture Detection is a computing process to recognize and interpret human gesture through a set of algorithms. Gesture Recognition functions by following these steps [1]:

Fig. 1. Gesture Recognition Process

1. Extraction Method and image pre-processing: The process of dividing the input image into regions separated by boundaries.
2. Features Extraction: Features vector of the segmented image to be extracted according to application.
3. Gestures Classification: Recognize the gesture

Gesture recognition has been implemented towards various field on different domain [1], including sign language translation, smart surveillance, virtual environment, etc.

C. Neural Network

An artificial neural network (ANN) is a computational model inspired by the structure and function of the human brain. ANNs are made up of small units called neurons, which are interconnected and communicate through synapses. These connections between neurons have weights that determine the strength of the signal being passed on. The neuron processes the information it receives from other neurons and produces an output that is then passed on to other neurons.

Types of Artificial Neural Networks:

1. Feedforward Artificial Neural Network: These networks have a unidirectional flow of information from input to output. The neurons in each layer are only connected to the neurons in the next layer.
2. Recurrent Artificial Neural Network: These networks have connections between neurons that allow information to flow back and forth. This allows the network to process sequential information, such as text or time signals.
3. Convolutional Neural Network: These networks are specifically designed to process spatial data, such as images or videos. These networks use convolution filters to extract features from spatial data.

D. Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of artificial neural network specifically designed to process spatial data, such as images or videos. CNNs use convolutional filters to extract features from spatial data. A convolution filter is a small matrix that is shifted across the input to calculate local features. These features are then combined to produce the output. CNN works by following these steps:

1. Input: Spatial data, such as images or videos, are fed into the CNN.
2. Convolution: Convolution filters are shifted across the input to calculate local features.
3. Pooling: The local features are combined to reduce the dimensionality of the output.
4. Fully Connected Layer: The extracted features are then connected to neurons in the fully connected layer.
5. Output: The fully connected layer produces output, such as image classification or object motion prediction.

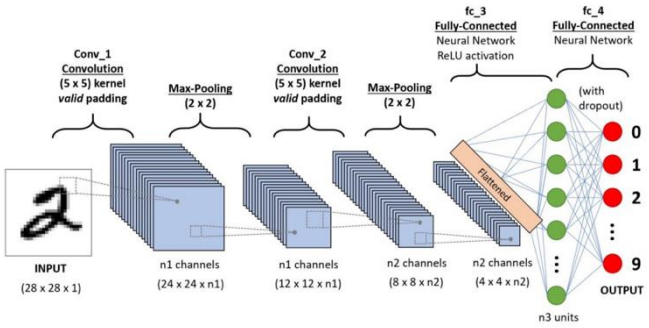


Fig. 2. CNN Process

CNNs are powerful for image analysis due to their ability to extract spatial features. This makes them ideal for tasks like image classification and object detection. CNNs can also be trained with relatively less data compared to other neural network architectures. However, CNNs can be complex and difficult to interpret due to their large number of parameters. Additionally, training CNNs often requires a substantial amount of labeled data, which can be expensive or limited in certain scenarios. CNN has been implemented toward various field such as:

1. Image Classification: CNNs excel at classifying objects in images. This benefits e-commerce (product image browsing) and self-driving cars (object recognition on the road).
2. Object Detection: CNNs pinpoint objects within images, crucial for self-driving cars (detecting pedestrians, vehicles, traffic lights) and security systems (identifying intruders).

3. Facial Recognition: CNNs are used for facial recognition in security, social media, and law enforcement.

III. Methodology

Flowchart of the experiment that will be conducted in this research is as below.

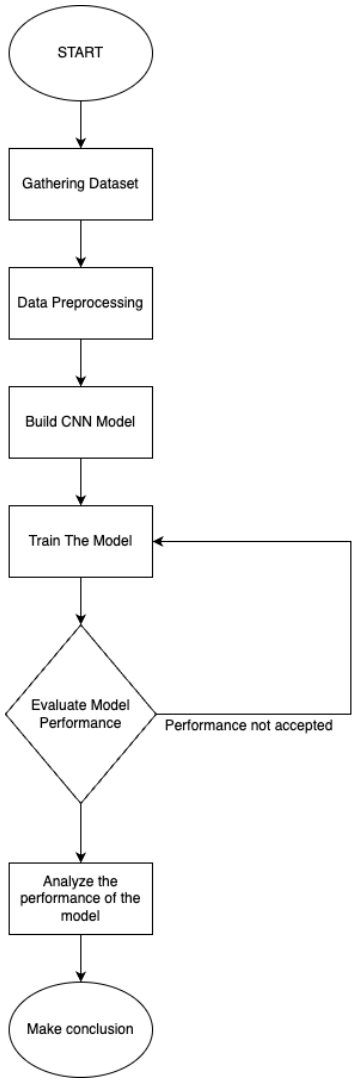
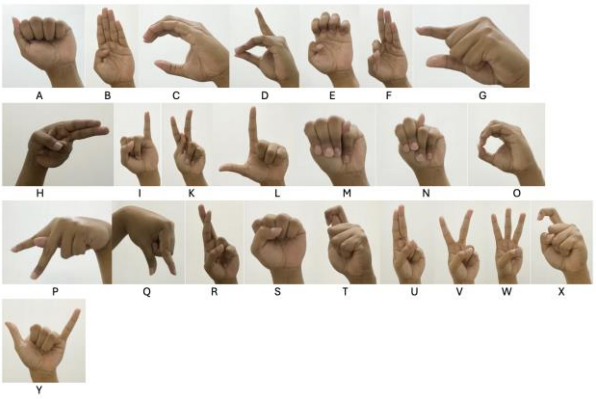


Fig. 3. Experiment Research Flow

A. Datasets

For our study, we utilized an open dataset of alphabets consisting of the letters A to Y, excluding J and Z due to their unique requirements in SIBI sign language. Example of image used in the dataset:



The dataset comprises 8,652 images categorized into 24 classes. We divided the dataset into three parts: Training (89%), Validation (10%), and Testing (1%).

B. Model Architecture

In this application, we utilize MobileNetV2, a lightweight and efficient convolutional neural network designed for mobile and edge devices, as the base of our model. MobileNetV2 features depth wise separable convolutions and an inverted residual structure, which significantly reduces the number of parameters and computational cost while maintaining high performance. The base model was initialized with pre-trained weights from ImageNet, and the initial layers were frozen to preserve these pre-trained weights. However, the last 50 layers were unfrozen to allow fine-tuning. On top of the base model, we added custom layers to adapt it for sign language translation:

- Global Average Pooling: To reduce the spatial dimensions.
- Batch Normalization: To stabilize and speed up the training process.
- Dense Layers: Two dense layers with ReLU activation functions and dropout for regularization.
- Output Layer: A dense layer with a softmax activation function to classify the 24 classes.

C. Dataset Preprocessing

We utilized ImageDataGenerator from TensorFlow to preprocess and augment our dataset. The training data was augmented with several transformations to enhance the model's robustness:

- Rescaling: Pixel values were rescaled to the range [0, 1].
- Rotation: Images were randomly rotated by up to 40 degrees.
- Shifting: Horizontal and vertical shifts up to 20% were applied.
- Shearing: Shear transformations up to 20% were applied.
- Zooming: Zoom transformations up to 20% were applied.
- Flipping: Images were horizontally flipped.

The validation and test data were only rescaled to the range [0, 1].:

D. Training Model

The training process was conducted over 150 epochs with a batch size of 64. We implemented several callbacks to enhance the training process:

- Early Stopping: Monitored the validation loss and stopped training if it did not improve for 10 consecutive epochs, restoring the best weights.
- Model Checkpoint: Saved the model's best weights based on the highest validation accuracy.
- ReduceLROnPlateau: Reduced the learning rate by a factor of 0.2 if the validation loss did not improve for 5 consecutive epochs, with a minimum learning rate of 0.00001.

During training, data augmentation and validation steps were performed, with steps per epoch and validation steps calculated based on the number of samples and batch size. The final trained model was saved for future use in mobile applications.

E. Application Architecture

The application will be developed in Flutter framework and utilize 'flutter_tflite.dart' as the package to utilize the tflite model. In which the application will take the image from the device's camera and translate the sign language in the image using the model.

F. Evaluation Methods

Evaluation for the sign language translation model is done using metrics of Training loss, Training accuracy, validation loss, and validation accuracy. We further evaluate this model through real-time usage using open-cv as well as an image picker and camera from the mobile device to see the performance of this model.

IV. Result and Discussion

After 50 epochs of training, the model exhibited robustness with consistently high accuracy and minimal loss, showcasing its effective learning from the training data and proficient reduction of prediction errors. The results from testing the model using OpenCV are as follows:

OpenCV Testing	
Alphabets	***Accuracy***
A	100%
B	100%
C	99%
D	100%
E	100%
F	100%
G	100%
H	98%
I	100%
K	100%
L	100%
M	100%
N	100%
O	100%
P	100%
Q	100%
R	100%
S	100%
T	99%
U	98%
V	100%
W	100%
X	100%
Y	98%

The results from testing the model using OpenCV demonstrate excellent performance, achieving an average accuracy of over 99.7%. This underscores the model's capability to provide accurate translations on an OpenCV-enabled laptop device.

Almost similar result found on the mobile device using image picker using the test dataset. The results follow:

Image Picker Testing	
Alphabets	***Accuracy***
A	100%
B	100%
C	100%
D	100%
E	100%
F	99%
G	100%
H	100%
I	100%
K	100%
L	100%
M	100%
N	100%
O	100%
P	100%
Q	100%
R	100%
S	100%
T	99%
U	100%
V	100%
W	100%

Image Picker Testing	
Alphabets	***Accuracy***
X	100%
Y	100%

The results from testing the model using Image Picker on mobile devices demonstrate outstanding performance, achieving an average accuracy of over 99.9%. This highlights the model's ability to accurately translate images directly on users' mobile devices.

However, the result when testing the model from the users camera seems lackluster, the result follows:

Camera Testing		
Expected	***Predicted***	***Accuracy***
A	A	95%
B	M	44%
C	G	43%
D	L	33%
E	E	39%
F	Y	23%
G	G	81%
H	L	48%
I	Y	97%
K	L	61%
L	L	99%
M	M	100%
N	N	87%
O	O	95%
P	K	75%
Q	L	71%
R	E	46%
S	N	44%
T	E	23%
U	A	87%
V	L	93%
W	L	85%
X	Y	89%
Y	Y	78%

Based on the results from testing the model using camera input, the model achieved varied levels of accuracy across different letters of the alphabet. The model demonstrated high accuracy for some letters, such as M (100%), L (99%), and Y (78%), indicating robust performance in correctly identifying these signs. However, for other letters like F (23%), T (23%), and S (44%), the accuracy was comparatively lower, suggesting areas where the model may benefit from further optimization or additional training data.

We hypnotize that the feature extraction on the data from the camera is not properly working and the model can't detect the sign language that Is being used. Therefore, further testing with improvement to the hand detection for this model is recommended to further improve this model.

V. Conclusion

The aim of this research is to develop a gesture recognition system that uses convolutional neural networks (CNN) to translate SIBI (Indonesian Sign System). The model shows excellent performance with an average accuracy of over 99.7% in OpenCV tests and over 99.9% in mobile device tests with image selector. However, when tested using a camera, accuracy varied, with some characters achieving high accuracy and others showing significant errors.

These findings indicate that although the model is highly effective in controlled environments, further optimization is needed for real-world applications, especially when improving feature extraction from camera input. Future work should focus on improving hand detection and expanding the dataset to increase the robustness and accuracy of the model under various conditions.

Overall, this research contributes to advances in SIBI sign language recognition technology and paves the way for communication tools that are more easily accessible to the deaf community in Indonesia.

References

[1] R. Zaman Khan, "Hand Gesture Recognition: A Literature Review," International Journal of Artificial Intelligence \& Applications, vol. 3, no. 4, pp. 161–174, Jul. 2012, doi: 10.5121/ijaia.2012.3412.

[2] Rafiqul Z. Khan, Noor A. Ibraheem, (2012). "Survey on Gesture Recognition for Hand Image Postures", International Journal of Computer And Information Science, Vol. 5(3), Doi: 10.5539/cis.v5n3p110

[3] M. Jonathan and E. Rakun, "Translating SIBI (Sign System for Indonesian Gesture) Gesture-to-Text in Real-Time using a Mobile Device", J. ICT Res. Appl., vol. 16, no. 3, pp. 259-280, Dec. 2022.

[4] T. Handhika, R. I. M. Zen, Murni, D. P. Lestari, and I. Sari, "Gesture recognition for Indonesian Sign Language (BISINDO)," Journal of Physics: Conference Series, vol. 1028, p. 012173, Jun. 2018, doi: 10.1088/1742-6596/1028/1/012173.

[5] S. Dwijayanti, H. -, S. I. Taqiyyah, H. Hikmarika, and B. Y. Suprapto, "Indonesia Sign Language Recognition using Convolutional Neural Network," International Journal of Advanced Computer Science and Applications, vol. 12, no. 10, 2021, doi: 10.14569/ijacsa.2021.0121046.

[6] N. Ahmad, E. S. Wijaya, C. Tjoaquinn, H. Lucky, and I. A. Iswanto, "Transforming Sign Language using CNN Approach based on BISINDO Dataset," 2023 International Conference on Informatics, Multimedia, Cyber and Informations System (ICIMCIS), Nov. 2023, Published, doi: 10.1109/icimcis60089.2023.10349011.

[7] A. R. Syulistyo, D. S. Hormansyah, and P. Y. Saputra, "SIBI (Sistem Isyarat Bahasa Indonesia) translation using Convolutional Neural Network (CNN)," IOP Conference Series: Materials Science and Engineering, vol. 732, no. 1, p. 012082, Jan. 2020, doi: 10.1088/1757-899x/732/1/012082.

[8] A. Anwar, A. Basuki, R. Sigit, A. Rahagiyanto, and Moh. Zikky, "Feature Extraction for Indonesian Sign Language (SIBI) Using Leap Motion Controller," 2017 21st International Computer Science and Engineering Conference (ICSEC), Nov. 2017, Published, doi: 10.1109/icsec.2017.8443926.

[9] I. Papastratis, C. Chatzikonstantinou, D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Artificial Intelligence Technologies for

Sign Language," Sensors, vol. 21, no. 17, p. 5843, Aug. 2021, doi: 10.3390/s21175843.

[10] D. Parashar, S. Thakur, K. B. Raju, G. B. Madhavi, and K. Sharma, "A Deep Learning-Based Approach for Hand Sign Recognition Using CNN Architecture," Revue d'Intelligence Artificielle, vol. 37, no. 4, pp. 937–943, Aug. 2023, doi: 10.18280/ria.370414.

[11] G. Strobel, T. Schoormann, L. Banh, and F. Möller, "Artificial Intelligence for Sign Language Translation – A Design Science Research Study," Communications of the Association for Information Systems, vol. 53, no. 1, pp. 42–64, 2023, doi: 10.17705/1cais.05303.

[12] Papatsimouli, Maria & Kollias, Konstantinos & Lazaridis, Lazaros & Maraslidis, George S. & Michailidis, Heracles & Sarigiannidis, Panagiotis & Fragulis, George. (2022). Real Time Sign Language Translation Systems: A review study. 1-4. 10.1109/MOCAST54814.2022.9837666.

[13] Darmawan, I Dewa Made Bayu Atmaja & Linawati, Linawati & Sukadarmika, Gede & Wirastuti, Ni & Pulungan, Reza & Mulyanto, Mulyanto & Hariyanti, Ni Kadek. (2023). Advancing Total Communication in SIBI: A Proposed Conceptual Framework for Sign Language Translation. 23-28. 10.1109/ICSGTEIS60500.2023.10424020.

[14] Saiful, Md & Isam, Abdulla & Moon, Hamim & Tammana, Rifa & Das, Mitul & Alam, Md & Rahman, Ashifur. (2022). Real-Time Sign Language Detection Using CNN. 10.1109/ICDABI56818.2022.10041711.

[15] A. Deshpande, A. Shriwas, V. Deshmukh and S. Kale, "Sign Language Recognition System using CNN," 2023 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bengaluru, India, 2023, pp. 906-911, doi: 10.1109/IITCEE57236.2023.10091051.