EfficientNet B0 Algorithm for SIBI to Text and Audio Translation System

A Muflih Zaizafuny[1], Rieke Adriati Wijayanti [2*], Sri Wahyuni Dali [3]

[1,2,3] Digital Telecommunication Network
Department of Electronic Engineering, Polytechnic State of Malang, Indonesia

[1]1841160105@polinema.ac.id, [2]riekeaw@polinema.ac.id, [3]sri.wahyuni@polinema.ac.id

Abstract— **The Indonesian Sign Language System (SIBI) serves as the fundamental mode of communication for the deaf and mute community in Indonesia. However, a significant communication gap exists as the general public often struggles to comprehend these signs. Leveraging advancements in Artificial Intelligence (AI), specifically Machine Learning, offers a solution to bridge this gap and improve accessibility. This research focuses on developing a SIBI sign language translation system that converts gestures into both text and audio formats, utilizing a Convolutional Neural Network (CNN) based on the EfficientNet B0 architecture. The model underwent rigorous training with varying epoch counts, ultimately identifying 35 epochs as the optimal duration to maximize performance while mitigating overfitting and underfitting risks. Experimental results from distance testing revealed an inverse relationship between accuracy and camera-to-hand distance: 74% at 5 cm, 64% at 25 cm, 52% at 45 cm, 41% at 55 cm, 28% at 65 cm, 26% at 75 cm, 16% at 85 cm, and 0% at 95 cm. These findings underscore distance as a critical factor. Overall, the application achieved an average accuracy of 76%, demonstrating its potential to facilitate effective communication and promote social inclusivity.**

Keywords*— *Sign Language System, Deep learning, Convolutional Neural Network, Pre-trained Model, Hyperparameter Tuning

I. INTRODUCTION

Deafness is a medical condition in which a person loses the ability to hear sounds or noises. Deafness can be partial or total, and it may occur from birth or as a result of health conditions or accidents. Several risk factors can cause deafness, including recurring ear infections, excessive exposure to loud noises, head injuries, or genetic factors. This condition can disrupt an individual's ability to communicate and interact with their environment, significantly impacting their overall quality of life [1].

According to information obtained from the Disability Information System of the Ministry of Health of the Republic of Indonesia in March 2022, the number of individuals with disabilities in Indonesia reached 212,240. This marks an increase over the past two years. In March 2020, the number of individuals with disabilities was recorded at 197,582, and in March 2021, the number rose to 207,604. As of March 2022, the number of individuals with speech and hearing disabilities was 19,392, equivalent to approximately 9.14% of the total number of individuals with disabilities in Indonesia [2].

Deaf individuals communicate using sign language, which involves hand gestures, facial expressions, or body movements to form symbols representing letters or words. In Indonesia, two popular sign languages are used: the Indonesian Sign Language System (SIBI) and the Indonesian Sign Language (BISINDO). However, SIBI is more often considered a learning system, while BISINDO emerges naturally from early-life interactions [3].

Previous research titled "The Application of Alphabet Recognition in the Indonesian Sign Language System (SIBI) Using the YOLOv5 Algorithm" aimed to develop a practical platform for recognizing the alphabet of the Indonesian Sign Language System (SIBI) as an effective medium for learning sign language. The method employed was the YOLOv5 algorithm to detect SIBI sign language gestures. Testing was conducted in two stages: first, to evaluate the application interface, which successfully displayed six user interface pages; and second, manual testing to detect SIBI sign language gestures by comparing the actual classes consisting of 26 categories [4].

In subsequent research titled "Implementation of Hand Sign Language Detection Using OpenCV and MediaPipe," the aim was to develop a system to facilitate direct learning of hand sign language. Sign language is a means of communication used by deaf individuals, typically involving body and lip movements. For those unfamiliar with sign language, confusion often arises, highlighting the need for a system to assist in learning it [5].

Therefore, an innovative solution is required to help individuals with speech and hearing disabilities communicate more easily and effectively. This thesis focuses on developing a SIBI-to-text-and-audio translator system using an integrated approach with the EfficientNet B0 Convolutional Neural Network (CNN) model to recognize SIBI hand gestures from images. The model will be integrated with a hand motion detection algorithm for conversion into text and audio. The primary goal of this research is to contribute to improving communication accessibility for individuals with hearing impairments through innovative and effective technology [6].

*Corresponding author
E-ISSN: 2654-6531 P- ISSN: 2407-0807

II. METHOD

In the development of a sign language alphabet identification system for SIBI using the CNN algorithm, several stages form the core of the development plan. This design aims to provide a comprehensive overview of the application's process or workflow, from start to finish. Fig. 1 is a flowchart that was created.

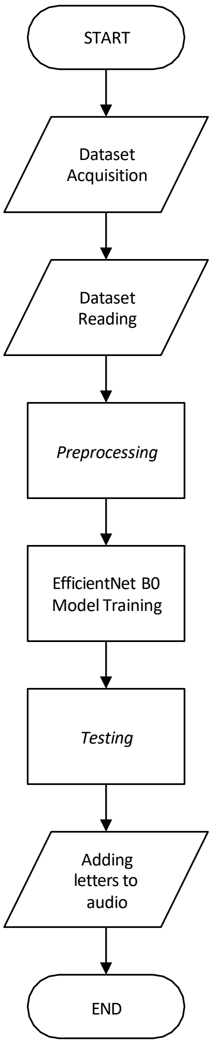


Figure 1. Workflow of Method Implementation

A. Dataset Acquisition

Collecting a dataset of hand gesture images representing the letters in the Indonesian Sign Language System (SIBI). The image capture process is carried out using a mobile phone camera to record hand gestures for each letter in SIBI. The dataset consists of hand gesture images, each taken for every letter of the SIBI alphabet, except for the letter 'Z' (which does not have a fixed gesture). After recording, these images are categorized and stored in folders according to the class or letter they represent [7].

B. Dataset Reading

At this stage, the system reads the dataset that has been previously collected. This dataset consists of 25 classes of hand gesture images, each representing a letter in the SIBI alphabet. These images will be used to train and test the gesture classification model. Dataset reading involves preparing the data for further processing in the subsequent stages, ensuring that each image is correctly associated with its corresponding letter label.

C. Preprocessing

Dataset preprocessing is conducted to prepare the data before being used for model training. The preprocessing steps include:

- Labeling: Each image is labeled according to the letter it represents.
- Image Resizing: All images are resized to a resolution of 96x96 pixels to match the input requirements of the EfficientNetB0 model [8]. This resolution is chosen for its efficiency in processing images with the CNN model without compromising gesture recognition quality.
- Dataset Splitting: The dataset is divided into three parts: 80% for training data, 10% for validation data, and 10% for testing data. This division ensures that the model has sufficient data for training while retaining data for performance evaluation.

D. EfficientNet B0 Model Training

In this stage, the EfficientNetB0 model is used to learn the patterns and characteristics of hand gestures in the dataset. This CNN model is chosen for its efficiency in processing image data with high accuracy while utilizing fewer computational resources compared to other models.

- Defining the Number of Classes: Setting 25 classes (for letters A-Y, excluding Z).
- Determining Training Parameters: Specifying parameters such as the number of epochs, batch size, and optimization functions.
- Training the Model: Using the training dataset to enable the model to learn the existing hand gesture patterns.
- Saving Model Parameters: Storing the trained model parameters for use in testing and classification processes.

E. Testing

After training is completed, the model is tested using the testing dataset prepared earlier (10% of the total dataset). This testing aims to evaluate the model's performance in recognizing hand gestures it has not encountered before. The testing is conducted by inputting hand gesture image data from the testing dataset into the model [9].

F. Adding Letters to Audio

Once the model recognizes the hand gesture and identifies the corresponding letter, the system will convert the letter into audio using the Google Cloud Speech API.

III. RESULTS AND DISCUSSION

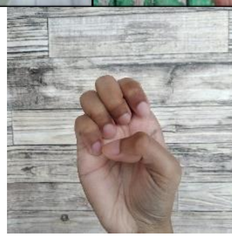
The results of the research should be written clearly and concisely. Discussions consider outlines the importance of research, not repeat it. Avoid excessive uses quotations and discussions about literature published.

A. Data Collection

In the data collection phase for the Indonesian Sign Language System (SIBI) translator system, images of hand gestures representing alphabet letters were gathered through direct photography and online sources. This process involved photographing each hand gesture individually to ensure the accuracy and consistency of each letter in the SIBI alphabet [10]. Additionally, some images were sourced from the internet to expand data diversity and address minor variations in hand gesture interpretations.

Data collection was conducted carefully to ensure all alphabet letters were well-represented, providing a clear visual representation of each gesture. The resulting images were then classified based on the letters they represent and stored in directories according to their respective classes, facilitating the training phase of the SIBI hand gesture recognition model. Examples of these images are as Table 1.

TABLE I
EXAMPLES OF DATA IMAGES

Word	Image
A	 
B	 
C	 
D	 
E	 

The data used in this study was obtained from the Kaggle platform, sourcing images across 25 classes (letters A-Y). This dataset was processed to retrain the image detection model. The data was split into 80% of the total samples for training data, 10% for validation data, and 10% for testing data [11]. The number of images is detailed in the Table 2.

TABLE II
NUMBER OF IMAGES IN THE DATASET

No.	Class	Total Images
1	A	603
2	B	603
3	C	595
4	D	464
5	E	589
6	F	588
7	G	520
8	H	602

No.	Class	Total Images
9	I	608
10	J	606
11	K	600
12	L	600
13	M	610
14	N	619
15	O	603
16	P	607
17	Q	612
18	R	607
19	S	612
20	T	608
21	U	609
22	V	610
23	W	605
24	X	596
25	Y	596

B. Data Augmentation

In the data augmentation process for SIBI sign language detection research, several modifications are applied to the original images to enhance data diversity and improve the model's ability to recognize variations in hand gestures. Augmentation is performed using Keras's ImageDataGenerator, enabling automated image processing with random transformations [12]. These transformations include rotating images up to 10 degrees, shifting the image position horizontally and vertically by 10%, zooming in or out by 10%, and adjusting image brightness within a range of 85% to 115% of the original value [13]. These modifications aim to introduce visual variations, ensuring the model does not rely solely on features from the original images but can recognize hand gestures under diverse conditions. Examples of augmented images are presented on Fig. 2:

Figure 2. Augmentation Results for a Specific Letter

This augmentation helps increase the amount of training data, which is essential to reducing the likelihood of overfitting in the model and providing better generalization when applied in real-world scenarios where lighting conditions, viewing angles, and hand positions can vary significantly. In the code, the number of augmented images is adjusted to meet the desired quantity, reaching 640 images for each class. Thus, in addition to the original data, the model is also trained using augmented data, which is expected to improve the model's accuracy and performance in recognizing SIBI sign language [14].

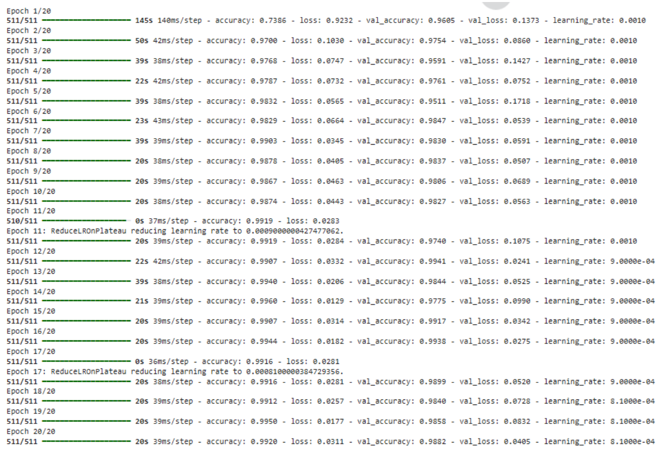
C. Model Training

The model training process in this study utilizes the EfficientNet B0 architecture. The dataset was downloaded from the Kaggle platform, converted to ZIP format, and uploaded to Google Drive for easier management. This processed dataset consists of 25 classes representing letters A to Y (excluding Z) as required for the study. The dataset was also augmented to enhance image diversity and subsequently imported into Google Colaboratory for training.

Images in the dataset serve as learning material for the model. The model's performance is evaluated based on its accuracy in predicting images learned during the training process and its ability to recognize new, unseen images. This evaluation ensures the model avoids both overfitting and underfitting. Overfitting occurs when the model predicts training data with high accuracy but performs poorly on new data. In contrast, underfitting happens when the model fails to make accurate predictions on both training and new data.

The model was trained with three variations in the number of epochs—20, 35, and 50—to assess its performance under different learning scenarios. A larger number of epochs allows the model to achieve higher performance but also increases the risk of overfitting. On the other hand, fewer epochs may result in suboptimal learning and a higher likelihood of underfitting. The training process used a batch size of 32 to ensure efficient data processing [15].

The choice of three variations in the number of epochs aims to evaluate how the model's performance changes with different epoch counts. By comparing the results of each epoch variation, researchers can determine the number of epochs that best balances accuracy on training data and generalization ability on new data, minimizing the risks of overfitting and underfitting. Results of model training are presented on Fig. 3.



```
Epoch 1/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 145s 140ms/step - accuracy: 0.7386 - loss: 0.9232 - val_accuracy: 0.9605 - val_loss: 0.1373 - learning_rate: 0.0010
Epoch 2/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 50s 42ms/step - accuracy: 0.9700 - loss: 0.1030 - val_accuracy: 0.9754 - val_loss: 0.0860 - learning_rate: 0.0010
Epoch 3/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 39s 38ms/step - accuracy: 0.9768 - loss: 0.0747 - val_accuracy: 0.9591 - val_loss: 0.1427 - learning_rate: 0.0010
Epoch 4/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 22s 42ms/step - accuracy: 0.9787 - loss: 0.0732 - val_accuracy: 0.9761 - val_loss: 0.0752 - learning_rate: 0.0010
Epoch 5/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 39s 38ms/step - accuracy: 0.9832 - loss: 0.0565 - val_accuracy: 0.9511 - val_loss: 0.1718 - learning_rate: 0.0010
Epoch 6/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 23s 43ms/step - accuracy: 0.9829 - loss: 0.0664 - val_accuracy: 0.9847 - val_loss: 0.0539 - learning_rate: 0.0010
Epoch 7/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 39s 39ms/step - accuracy: 0.9903 - loss: 0.0345 - val_accuracy: 0.9830 - val_loss: 0.0591 - learning_rate: 0.0010
Epoch 8/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 38ms/step - accuracy: 0.9878 - loss: 0.0405 - val_accuracy: 0.9837 - val_loss: 0.0507 - learning_rate: 0.0010
Epoch 9/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 39ms/step - accuracy: 0.9867 - loss: 0.0463 - val_accuracy: 0.9806 - val_loss: 0.0689 - learning_rate: 0.0010
Epoch 10/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 38ms/step - accuracy: 0.9874 - loss: 0.0443 - val_accuracy: 0.9827 - val_loss: 0.0563 - learning_rate: 0.0010
Epoch 11/20
510/511 ━━━━━━━━━━━━━━━━━━━ 0s 37ms/step - accuracy: 0.9919 - loss: 0.0283
Epoch 11: ReduceLROnPlateau reducing learning rate to 0.0009000000427477062.
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 39ms/step - accuracy: 0.9919 - loss: 0.0284 - val_accuracy: 0.9740 - val_loss: 0.1075 - learning_rate: 0.0010
Epoch 12/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 22s 42ms/step - accuracy: 0.9907 - loss: 0.0332 - val_accuracy: 0.9941 - val_loss: 0.0241 - learning_rate: 9.0000e-04
Epoch 13/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 39s 38ms/step - accuracy: 0.9940 - loss: 0.0206 - val_accuracy: 0.9844 - val_loss: 0.0525 - learning_rate: 9.0000e-04
Epoch 14/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 21s 39ms/step - accuracy: 0.9960 - loss: 0.0129 - val_accuracy: 0.9775 - val_loss: 0.0990 - learning_rate: 9.0000e-04
Epoch 15/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 39ms/step - accuracy: 0.9907 - loss: 0.0314 - val_accuracy: 0.9917 - val_loss: 0.0342 - learning_rate: 9.0000e-04
Epoch 16/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 39ms/step - accuracy: 0.9944 - loss: 0.0182 - val_accuracy: 0.9938 - val_loss: 0.0275 - learning_rate: 9.0000e-04
Epoch 17/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - accuracy: 0.9916 - loss: 0.0281
Epoch 17: ReduceLROnPlateau reducing learning rate to 0.0008100000384729356.
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 38ms/step - accuracy: 0.9916 - loss: 0.0281 - val_accuracy: 0.9899 - val_loss: 0.0520 - learning_rate: 9.0000e-04
Epoch 18/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 39ms/step - accuracy: 0.9912 - loss: 0.0257 - val_accuracy: 0.9840 - val_loss: 0.0728 - learning_rate: 8.1000e-04
Epoch 19/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 39ms/step - accuracy: 0.9950 - loss: 0.0177 - val_accuracy: 0.9858 - val_loss: 0.0832 - learning_rate: 8.1000e-04
Epoch 20/20
511/511 ━━━━━━━━━━━━━━━━━━━━ 20s 39ms/step - accuracy: 0.9920 - loss: 0.0311 - val_accuracy: 0.9882 - val_loss: 0.0405 - learning_rate: 8.1000e-04
```

Figure 3. Model Training

D. Training Results Graph

- Epoch 20

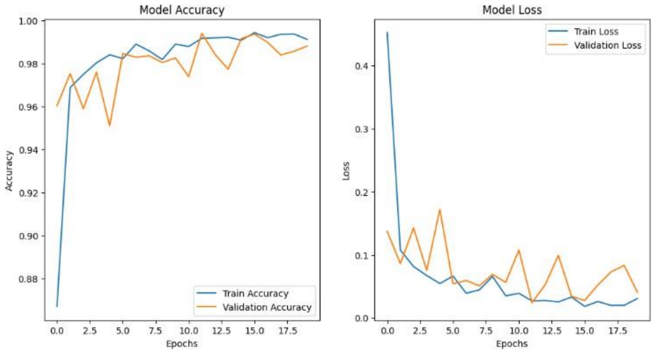


Figure 4.Train with 20 epochs

In the training graph from Fig. 4 with 20 epochs, the model demonstrates reasonably good performance with high accuracy on both training and validation data, as well as stable loss values. However, there are indications that the model has not learned optimally, as the difference between train accuracy and validation accuracy remains slightly fluctuating, and the loss value has not fully stabilized. This suggests the possibility of slight underfitting, where the model has not completely captured the patterns in the training data.

- Epoch 35

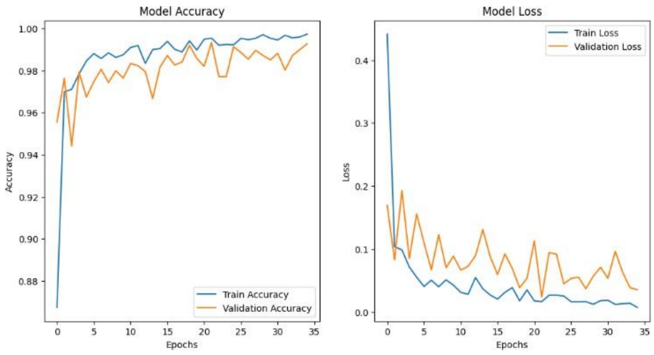


Figure 5.Train with 35 epochs

In the training graph from Fig. 5 with 35 epochs, the model demonstrates its best performance. The training and validation accuracy are consistently high, with a minimal gap between them, indicating that the model has good generalization ability. Additionally, the train loss and validation loss values are stable and exhibit nearly parallel patterns, suggesting that the model is in an optimal state without significant indications of overfitting or underfitting.

- Epoch 50

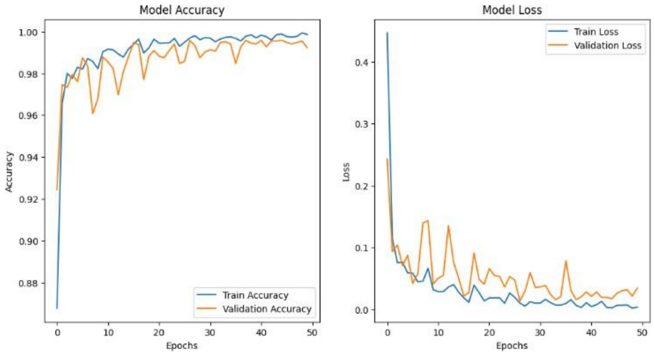


Figure 6.Train with 50 epochs

In the training graph from Fig. 6 with 50 epochs, the model's performance begins to show signs of overfitting. The accuracy on the training data is very high, but the validation accuracy is slightly lower, with a noticeable gap emerging between the two. The validation loss also exhibits greater fluctuations compared to the training loss, further indicating that the model is overly focused on the training data and losing its ability to generalize to new data.

Based on this analysis, it can be concluded that training with 35 epochs provides the most optimal results, achieving the best balance between accuracy and generalization ability without experiencing overfitting or underfitting.

E. Confusion Matrix

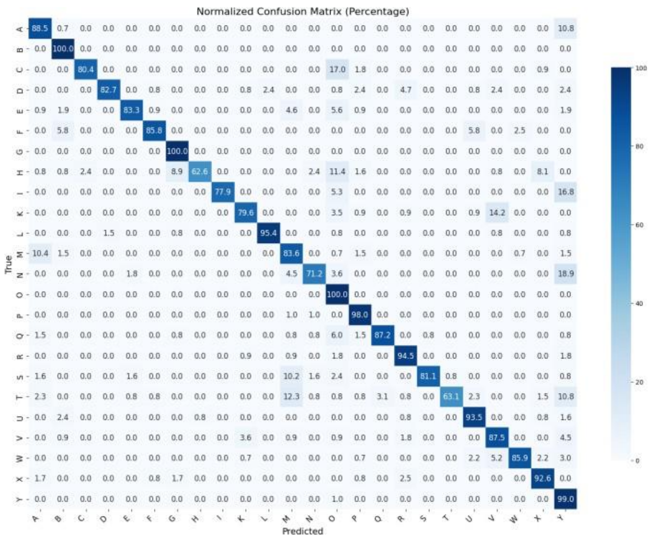


Figure 7.Confusion Matrix

Based on the Fig. 7 normalized confusion matrix, the model demonstrates strong performance in recognizing certain letters such as \`C\`, \`G\`, \`M\`, \`O\`, and \`P\`, with accuracy nearing or reaching 100%. However, there are some letters with lower accuracy rates, such as \`H\` (62.6%), \`T\` (63.1%), and \`S\` (81.1%), which are often misclassified as other letters. For instance, \`H\` is frequently identified as \`I\` (14.2%) or \`G\` (11.4%), while \`T\` is commonly misinterpreted as \`S\` (12.3%) or \`W\` (10.8%). These errors are likely caused by visual feature similarities between these letters, noise in the data, or variations in hand poses.

F. Application on Android

This Android application was developed using the Java programming language, one of the most widely used languages for Android application development. The interface design of the application adopts a minimalist and straightforward approach. The goal of this design choice is to ensure that users can easily understand and operate the application. Therefore, the application features a single main page interface. This allows users to access the functionality directly without the need to switch between screens. Below is the layout of the application's main page as designed on Fig. 8:

Figure 8. Initial Interface Image

The image shows the main page layout of the designed application. On this main page, there is a "Start" button to initiate the SIBI Sign Language Detection application.

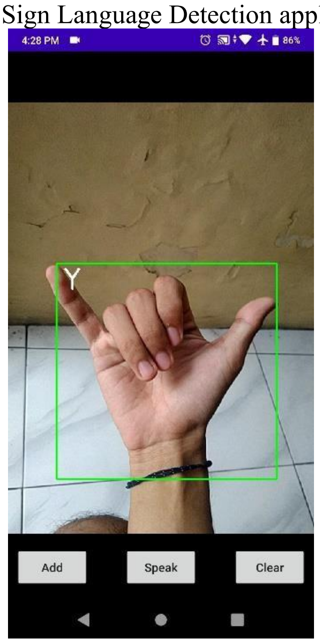


Figure 9. Output Page Display Image

The image from Fig. 9 illustrates the output page layout of the designed application. After the user performs a SIBI sign language gesture for a letter, the image is processed by the model for classification. The prediction results from the classification are then displayed on the same page. On the output page, the prediction result is shown in white text indicating the detected letter. For example, in the image, the user performs the SIBI gesture for the letter “Y.” This design allows users to view the classification results directly, with clear and easily understandable information.

G. Alphabet Recognition Distance Testing

To evaluate how well the system can recognize sign language at various distances, tests were conducted at eight distance ranges: 5 cm, 25 cm, 45 cm, 55 cm, 65 cm, 75 cm, 85 cm, and 95 cm. This test aims to analyze whether the distance between the hand and the camera affects detection accuracy. Each sign was tested five times at each distance, and the results were compared to determine if there was a pattern of accuracy decline as the distance increased. Below are the test results for distances of 5 cm, 25 cm, 45 cm, 55 cm, 65 cm, 75 cm, 85 cm, and 95 cm.

- 5 cm (centimeter)

Testing at a distance of 5 cm was conducted to evaluate the system's performance in recognizing sign language at an extremely close range. Fig. 10 below is an example of detection at a distance of 5 cm:

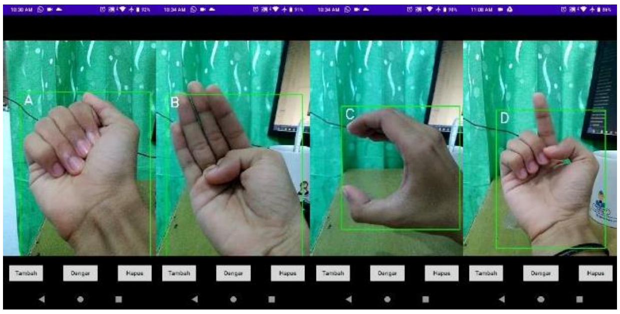


Figure 10. Example of Detection Results at 5 cm

In the 5 cm distance test, detection was performed five times for each class. The results are presented in Table 3.

TABLE III
TESTING RESULTS AT 5 CM

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
A	√	√	√	√	√	100%
B	√	√	√	√	√	100%
C	√	√	√	√	√	100%
D	√	√	X	√	√	80%
E	X	√	√	√	√	80%
F	√	√	√	√	√	100%
G	√	X	√	√	X	60%
H	√	X	√	X	X	40%
I	X	√	X	√	√	60%
J	X	√	X	√	√	60%
K	√	√	√	√	√	100%
L	√	√	√	√	√	100%
M	X	X	X	√	X	20%
N	√	X	X	X	√	40%
O	√	√	√	√	√	100%
P	X	X	X	X	√	20%
Q	√	√	√	√	√	100%
R	√	X	√	√	√	80%
S	√	X	√	√	X	60%
T	X	√	√	√	√	80%
U	√	√	√	X	√	80%
V	√	√	√	√	X	80%
W	√	√	√	√	√	100%
X	√	X	X	X	X	20%

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
Y	√	√	√	√	√	100%
Average Accuracy Value						**74%**

Testing at a distance of 5 cm was conducted to evaluate the system's performance in recognizing sign language at an extremely close range. The test results showed that the system achieved an accuracy of 74%, indicating that at this distance, the camera could capture gesture details fairly well.

- 25 cm (centimeter)

Testing at a distance of 25 cm was conducted to evaluate the system's performance under more common usage conditions. Fig. 11 below is an example of detection at a distance of 25 cm:

Figure 11. Example of Detection Results at 25 cm

In the 25 cm distance test, detection was performed five times for each class. The results are presented in Table 4.

TABLE IV
TESTING RESULTS AT 25 CM

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
A	√	√	√	X	√	80%
B	√	√	√	√	√	100%
C	√	√	√	√	√	100%
D	√	X	√	X	√	60%
E	X	√	X	√	√	60%
F	√	√	√	√	√	100%
G	X	√	X	√	X	40%
H	X	X	√	X	X	20%
I	X	√	X	√	X	40%
J	X	X	X	√	√	40%
K	√	√	√	√	√	100%
L	√	√	√	√	√	100%
M	X	X	X	√	X	20%
N	√	X	X	X	X	20%
O	√	√	√	√	√	100%
P	X	X	X	X	√	20%
Q	√	√	√	√	√	100%
R	√	X	√	X	√	60%
S	X	X	√	√	X	40%
T	√	√	X	X	√	60%
U	√	X	√	√	√	80%
V	√	√	√	√	X	80%
W	√	√	X	√	√	80%
X	X	X	X	X	X	0%
Y	√	√	√	√	√	100%
Average Accuracy Value						**64%**

Testing at a distance of 25 cm was conducted to evaluate the system's performance under more typical usage conditions. The test results showed that the system achieved an accuracy of 64%, which is lower compared to the 5 cm distance. This decrease in accuracy may be due to the reduced level of gesture detail captured by the camera or variations in hand positioning that are more difficult to recognize at a medium distance.

- 45 cm (centimeter)

Testing at a distance of 45 cm was conducted to evaluate the system's performance under more common usage conditions. Fig. 12 below is an example of detection at a distance of 45 cm:

Figure 12. Example of Detection Results at 45 cm

In the 45 cm distance test, detection was performed five times for each class. The results are presented in Table 5:

TABLE V
TESTING RESULTS AT 45 CM

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
A	√	X	X	√	√	60%
B	√	√	√	√	√	100%
C	√	√	√	√	√	100%
D	√	X	X	√	X	40%
E	X	√	X	√	√	60%
F	√	X	√	√	√	80%
G	X	X	X	X	√	20%
H	X	X	X	X	X	0%
I	X	√	X	X	X	20%
J	X	X	X	√	X	20%
K	√	√	√	√	X	80%
L	√	√	√	√	√	100%
M	X	X	X	X	X	0%
N	X	X	X	√	X	20%
O	√	√	√	√	√	100%
P	X	X	X	X	X	0%

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
Q	X	√	√	√	√	80%
R	X	X	√	X	√	40%
S	X	X	X	X	√	20%
T	X	X	√	X	√	40%
U	√	X	√	√	X	60%
V	√	√	X	√	√	80%
W	X	√	√	√	√	80%
X	X	X	X	X	X	0%
Y	√	√	√	√	√	100%
Average Accuracy Value						**52%**

Testing at a distance of 45 cm was conducted to analyze the system's ability to recognize sign language from a greater distance. The test results showed that the system achieved an accuracy of 52%, which is a decline compared to the 25 cm and 5 cm distances. This decrease in accuracy is likely due to the reduced gesture details captured by the camera, as well as an increased risk of background interference or more challenging hand positioning recognition.

- 55 cm (centimeter)

Testing at a distance of 55 cm was conducted to evaluate the system's performance in recognizing sign language from a farther distance. Fig. 13 below is an example of detection at a distance of 55 cm:

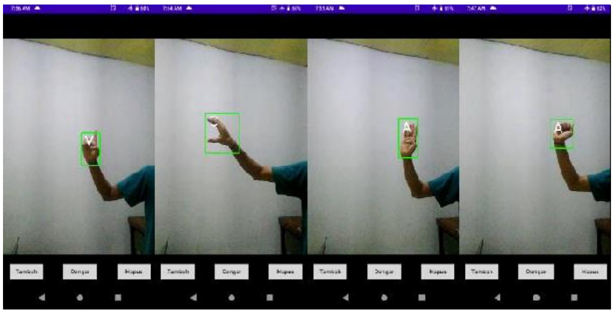


Figure 13. Example of Detection Results at 55 cm

In the 55 cm distance test, detection was performed five times for each class. The results are presented in Table 6.

TABLE VI
TESTING RESULTS AT 55 CM

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
A	X	X	X	√	X	20%
B	√	X	X	√	√	60%
C	√	√	X	X	√	60%
D	√	X	X	√	X	40%
E	X	√	X	√	√	60%
F	√	X	√	√	X	60%
G	X	X	X	X	√	20%
H	X	X	X	X	X	0%
I	X	√	X	X	X	20%
J	X	X	X	√	X	20%
K	√	√	√	√	X	80%
L	√	X	√	√	X	60%
M	X	X	X	X	X	0%
N	X	X	X	√	X	20%
O	√	X	√	X	√	60%
P	X	X	X	X	X	0%
Q	X	√	X	√	√	60%
R	X	X	√	X	X	20%
S	X	X	X	X	√	20%
T	X	X	√	X	√	40%
U	√	X	√	√	X	60%
V	X	√	X	√	√	60%
W	X	√	√	√	√	80%
X	X	X	X	X	X	0%
Y	√	√	√	√	√	100%
Average Accuracy Value						**41%**

Testing at a distance of 55 cm was conducted to analyze the system's ability to recognize sign language from a greater distance. The test results showed that the system achieved an accuracy of 41%, which is a decline compared to the 45 cm, 25 cm, and 5 cm distances. This decrease in accuracy is likely due to the reduced gesture details captured by the camera and the smaller hand size within the frame, making it more challenging for the system to recognize patterns with high accuracy. These results indicate that increasing the distance can affect the system's effectiveness in optimally detecting sign language.

- 65 cm (centimeter)

Testing at a distance of 65 cm was conducted to evaluate the system's performance in recognizing sign language from a farther distance. Fig. 14 below is an example of detection at a distance of 65 cm:

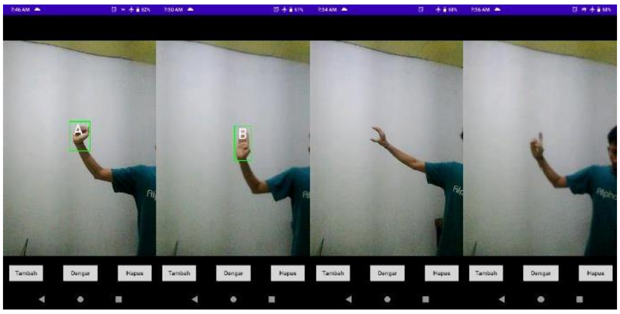


Figure 14. Example of Detection Results at 65 cm

In the 65 cm distance test, detection was performed five times for each class. The results are presented in Table 7.

TABLE VII
TESTING RESULTS AT 65 CM

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
A	X	X	X	√	X	20%
B	√	X	X	√	√	60%
C	√	X	X	X	√	40%

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
D	X	X	X	√	X	20%
E	X	X	X	√	X	20%
F	√	X	√	X	X	40%
G	X	X	X	X	X	0%
H	X	X	X	X	X	0%
I	X	√	X	X	X	20%
J	X	X	X	√	X	20%
K	X	X	X	√	X	20%
L	√	X	√	X	X	40%
M	X	X	X	X	X	0%
N	X	X	X	√	X	20%
O	√	X	√	X	√	60%
P	X	X	X	X	X	0%
Q	X	X	X	√	X	20%
R	X	X	√	X	X	20%
S	X	X	X	X	√	20%
T	X	X	√	X	X	20%
U	√	X	X	√	X	40%
V	X	√	X	√	√	60%
W	X	√	X	√	X	40%
X	X	X	X	X	X	0%
Y	√	√	√	√	√	100%
Average Accuracy Value						**28%**

Testing at a distance of 65 cm was conducted to evaluate the system's ability to recognize sign language from a greater distance. The test results showed that the system achieved an accuracy of 28%, which is a decline compared to the 55 cm, 45 cm, 25 cm, and 5 cm distances. This decrease in accuracy is likely due to the further reduction in gesture details captured by the camera and the smaller hand size within the frame, making it more difficult for the system to recognize patterns effectively. These results indicate that at this distance, the system begins to experience more significant challenges in accurately detecting sign language.

- 75 cm (centimeter)

Testing at a distance of 75 cm was conducted to evaluate the system's performance in recognizing sign language from a farther distance. Fig. 15 below is an example of detection at a distance of 75 cm:

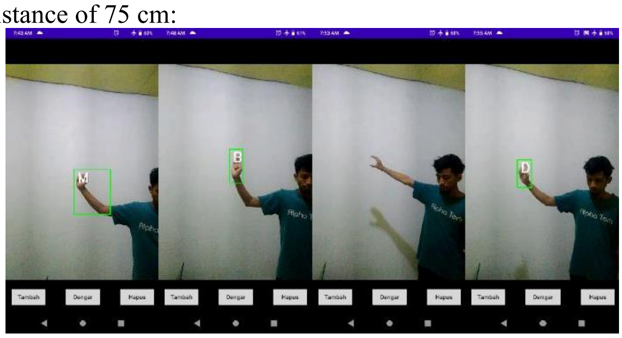


Figure 15. Example of Detection Results at 75 cm

In the 75 cm distance test, detection was performed five times for each class. The results are presented in Table 8.

TABLE VIII
TESTING RESULTS AT 75 CM

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
A	√	X	X	X	X	20%
B	X	√	X	√	√	60%
C	√	X	X	X	√	40%
D	X	X	X	X	X	0%
E	X	√	X	√	X	40%
F	X	X	X	√	X	20%
G	X	X	X	X	√	20%
H	X	X	X	X	X	0%
I	X	√	X	X	X	20%
J	X	X	X	√	X	20%
K	X	X	√	X	X	20%
L	√	X	√	X	X	40%
M	X	X	X	X	X	0%
N	X	X	X	√	X	20%
O	√	√	X	√	X	60%
P	X	X	X	X	X	0%
Q	X	X	X	X	√	20%
R	X	X	X	X	X	0%
S	X	X	X	X	√	20%
T	X	X	X	X	X	0%
U	√	X	√	X	X	40%
V	X	√	X	√	X	40%
W	X	X	√	√	√	60%
X	X	X	X	X	X	0%
Y	√	X	√	√	√	80%
Average Accuracy Value						**26%**

Testing at a distance of 75 cm was conducted to evaluate the system's ability to recognize sign language from a greater distance. The test results showed that the system achieved an accuracy of 26%, which is a decline compared to the 65 cm, 55 cm, 45 cm, 25 cm, and 5 cm distances. This decrease is likely due to the further reduction in gesture details captured by the camera and the smaller hand size within the frame, making it increasingly difficult for the system to recognize patterns accurately. These results indicate that at this distance, the system faces significant limitations in detecting sign language accurately.

- 85 cm (centimeter)

Testing at a distance of 85 cm was conducted to evaluate the system's performance in recognizing sign language from a farther distance. Fig. 16 below is an example of detection at a distance of 85 cm:

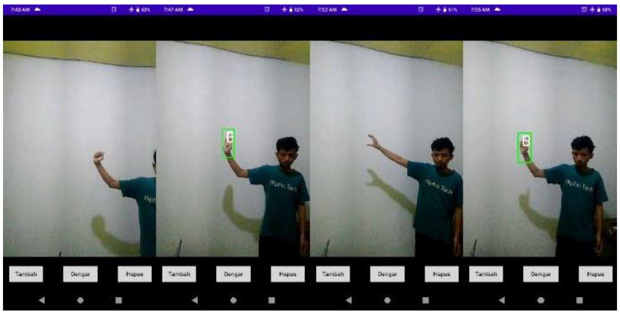


Figure 16. Example of Detection Results at 85 cm

In the 85 cm distance test, detection was performed five times for each class. The results are presented in Table 9.

TABLE IX
TESTING RESULTS AT 85 CM

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
A	X	√	√	X	X	40%
B	√	√	X	√	X	60%
C	X	X	√	X	X	20%
D	X	X	X	X	X	0%
E	X	X	X	√	X	20%
F	X	X	X	X	X	0%
G	X	X	X	X	X	0%
H	X	X	X	X	X	0%
I	X	√	X	X	√	40%
J	X	X	√	X	X	20%
K	X	X	X	X	X	0%
L	X	X	√	X	√	40%
M	X	X	X	X	X	0%
N	X	X	X	√	X	20%
O	X	X	√	X	√	40%
P	X	X	X	X	X	0%
Q	X	X	X	X	X	0%
R	X	X	X	X	X	0%
S	X	X	X	X	X	0%
T	X	X	X	X	X	0%
U	X	X	X	X	X	0%
V	X	√	X	X	X	20%
W	X	X	X	√	X	20%
X	X	X	X	X	X	0%
Y	√	X	√	X	√	60%
Average Accuracy Value						**16%**

Testing at a distance of 85 cm was conducted to evaluate the system's ability to recognize sign language from a greater distance. The test results showed that the system achieved an accuracy of 16%, which is a decline compared to the 75 cm, 65 cm, 55 cm, 45 cm, 25 cm, and 5 cm distances. This decrease is likely due to the smaller hand size within the frame and the reduced gesture details that the system can recognize. At this distance, the system exhibits greater difficulty in detecting patterns with high accuracy, indicating that its effectiveness in recognizing sign language continues to decline as the distance increases.

- 95 cm (centimeter)

Testing at a distance of 95 cm was conducted to evaluate the system's performance in recognizing sign language from a farther distance. Fig. 17 below is an example of detection at a distance of 95 cm:

Figure 17. Example of Detection Results at 95 cm

In the 95 cm distance test, detection was performed five times for each class. The results are presented in Table 10.

TABLE X
TESTING RESULTS AT 95 CM

Label	The number of testing stages					Accuracy
	1	2	3	4	5	
A	X	X	X	X	X	0%
B	X	X	X	X	X	0%
C	X	X	X	X	X	0%
D	X	X	X	X	X	0%
E	X	X	X	X	X	0%
F	X	X	X	X	X	0%
G	X	X	X	X	X	0%
H	X	X	X	X	X	0%
I	X	X	X	X	X	0%
J	X	X	X	X	X	0%
K	X	X	X	X	X	0%
L	X	X	X	X	X	0%
M	X	X	X	X	X	0%
N	X	X	X	X	X	0%
O	X	X	X	X	X	0%
P	X	X	X	X	X	0%
Q	X	X	X	X	X	0%
R	X	X	X	X	X	0%
S	X	X	X	X	X	0%
T	X	X	X	X	X	0%
U	X	X	X	X	X	0%
V	X	X	X	X	X	0%
W	X	X	X	X	X	0%
X	X	X	X	X	X	0%
Y	X	X	X	X	X	0%
Average Accuracy Value						**0%**

Testing at a distance of 95 cm was conducted to evaluate the system's limit in recognizing sign language from a very far distance. The test results showed that the system achieved an accuracy of 0%, representing a drastic decline compared to the 85 cm, 75 cm, 65 cm, 55 cm, 45 cm, 25 cm, and 5 cm distances. The system's failure at this distance is likely due to the hand being too small within the frame and the significant loss of gesture details that the model can recognize. At this distance, the system loses its ability to accurately detect sign language, indicating that there is a maximum effective range for the system's usability.

H. Accuracy Testing of Alphabet Recognition

The sign language model detection testing was conducted manually by using the application directly. The testing process involved pointing the phone's camera from the application toward the SIBI sign performer. Fig. 18 below is an example of a successful detection result:

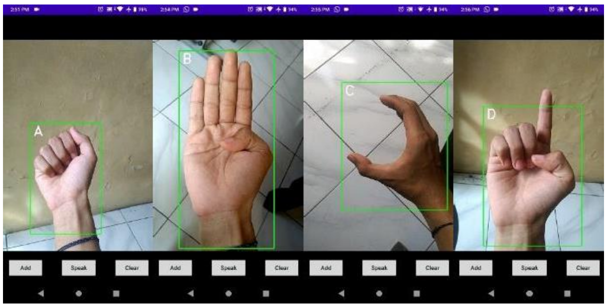


Figure 18. Example of Gesture Detection Results

In the accuracy testing, results were obtained from 10 detection attempts for each class. The outcomes are presented in the Table 11.

TABLE XI
ALPHABET TESTING TABLE

Label	The number of testing stages										Accuracy
	1	2	3	4	5	6	7	8	9	10	
A	√	√	√	√	√	√	√	√	√	√	100%
B	√	√	√	√	√	√	√	√	√	√	100%
C	√	√	√	√	√	√	√	√	√	√	100%
D	√	√	√	√	√	√	X	√	√	√	90%
E	√	X	√	X	√	√	√	√	√	√	80%
F	√	√	√	√	√	√	√	√	√	√	100%
G	√	√	X	√	√	X	√	X	√	√	70%
H	√	X	√	X	X	X	√	√	X	√	50%
I	X	√	X	X	√	√	X	√	√	√	60%
J	√	X	√	X	√	X	√	√	√	X	60%
K	√	√	√	√	√	√	√	√	√	√	100%
L	√	√	√	√	√	√	X	√	√	√	90%
M	X	X	√	√	X	√	X	X	X	X	30%
N	√	√	X	√	√	X	√	√	√	X	70%
O	√	√	√	√	√	√	√	√	√	√	100%
P	X	X	X	X	X	X	√	X	X	X	10%
Q	√	√	√	√	√	√	√	√	√	√	100%
R	√	√	√	√	√	X	√	X	√	√	80%
S	√	X	√	√	X	√	√	X	√	X	60%
T	X	√	√	√	√	√	√	√	X	√	80%
U	√	√	√	X	√	√	X	√	√	√	80%
V	√	X	√	√	√	√	√	√	√	√	90%
W	√	√	√	√	√	√	√	√	√	√	100%
X	√	X	X	X	X	√	X	√	X	X	30%
Y	√	√	√	√	√	√	√	√	√	√	100%
Average Accuracy Value											**76%**

Based on the test results table, the SIBI sign language detection system demonstrates varying performance levels. The label ‘M’ shows a low accuracy of 30%, due to its similarity to the pattern of the label ‘N’. Similarly, the label ‘P’ exhibits very low performance with an accuracy of only 10%, potentially caused by the complexity or the need for specific angles to correctly recognize its gesture.

On the other hand, several classes achieved perfect accuracy (100%), including ‘A', 'B', 'C', 'F', 'K', 'O', 'Q', 'W', and 'Y'. This indicates that the gestures for these classes have consistent patterns, are not too similar to other classes, and are easily recognized by the system. Classes such as 'D', 'L', and 'V' also performed very well, each with an accuracy of 90%.

Overall, the system achieved an average accuracy of 76%, reflecting fairly good performance.

IV. CONCLUSION

The model demonstrated optimal performance during training with 35 epochs. At this number of epochs, the model successfully balanced accuracy on the training data and generalization ability on the validation data without experiencing overfitting or underfitting. Training with 20 epochs resulted in good performance but showed signs of underfitting, while training with 50 epochs increased the risk of overfitting.

In the alphabet recognition distance test conducted at 5 cm, 25 cm, 45 cm, 55 cm, 65 cm, 75 cm, 85 cm, and 95 cm, the system's accuracy decreased as the distance increased, with respective accuracies of 74%, 64%, 52%, 41%, 28%, 26%, 16%, and 0%. The highest accuracy at 5 cm indicates that the camera can capture gesture details more clearly, whereas the decline in accuracy from 25 cm to 85 cm was due to reduced captured details and changes in hand size. At 95 cm, the system failed to recognize gestures entirely, likely because the hand appeared too small in the frame and contained insufficient visual information for the model to process. These results suggest that the system is more effective at close to medium distances but loses accuracy significantly at greater distances.

Overall, the system achieved an average accuracy of 76%, indicating that the application is feasible for further development and implementation based on testing conducted by evaluators familiar with basic sign language movements.

REFERENCES

[1] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, “Multiclass skin cancer classification using EfficientNets

– a first step towards preventing skin cancer," *Neurosci. Informatics*, vol. 2, no. 4, p. 100034, 2022, doi: 10.1016/j.neuri.2021.100034.

[2] A. Ma'aruf and M. Hardjianto, "Application of the You Only Look Once Version 8 Algorithm for Indonesian Sign Language Alphabet," *Semin. Nas. Mhs. Fak. Teknol. Inf.*, vol. 2, no. September, pp. 567–576, 2023.

[3] M. Zikky and Z. F. Akbar, "Kamus Sistem Isyarat Bahasa Indonesia (KASIBI) dengan Voice Recognition sebagai Pendukung Belajar Bahasa Isyarat Berbasis Android," *JST (Jurnal Sains Ter.*, vol. 5, no. 2, 2019, doi: 10.32487/jst.v5i2.732.

[4] D. Permana and J. Sutopo, "Aplikasi Pengenalan Abjad Sistem Isyarat Bahasa Indonesia (SIBI) Dengan Algoritma Yolov5" *J. Simantec*, vol. 11, no. 2, pp. 231–240, 2023, doi: 10.21107/simantec.v11i2.19783.

[5] A. R. Ardiansyah, A. H. Nur'azizan, and R. Fernandis, "Implementasi Deteksi Bahasa Isyarat Tangan Menggunakan OpenCV dan MediaPipe," *Stain. (Seminar Nas. Teknol. Sains)*, vol. 3, no. 1, pp. 331–337, 2024.

[6] P. A. Nugroho, I. Fenriana, R. Arijanto, and M. Kom, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia", ALGOR, Vol. 1, 2020.

[7] I. W. Suartika E. P., A. Y. Wijaya, and R. Soelaiman, "Klasifikasi citra menggunakan convolutional neural network (CNN) pada Caltech 101," *Jurnal Teknik ITS*, vol. 5, no. 1, 2016.

[8] A. Rohim, Y. A. Sari, and Tibyani, "Convolution neural network (CNN) untuk pengklasifikasian citra makanan tradisional," *JPTIIK*, vol. 3, no. 7, pp. 7037–7042, 2019.

[9] Q. Zheng, M. Yang, X. Tian, X. Wang, and D. Wang, "Rethinking the role of activation functions in deep convolutional neural networks for image classification," *Engineering Letters*, vol. 28, no. 1, pp. 80–92, 2020.

[10] T. M. Kadarina and M. H. I. Hajar, "Pengenalan bahasa pemrograman Python menggunakan aplikasi games untuk siswa/i di wilayah Kembangan Utara," *JAM: Jurnal Abdi Masyarakat* (Univ. Mercu Buana), pp. 11–16, 2019.

[11] G. I. E. Soen, Marlina, and Renny, "Implementasi cloud computing dengan Google Colaboratory pada aplikasi pengolah data Zoom participants," *JITU: Journal Informatic Technology and Communication*, vol. 6, no. 1, pp. 24–30, Jun. 2022.

[12] N. Wiranda, H. S. Purba, and R. A. Sukmawati, "Survei penggunaan TensorFlow pada machine learning untuk identifikasi ikan kawasan lahan basah," *IJEIS*, vol. 10, no. 2, pp. 179–188, 2020, doi: 10.22146/ijeis.58315.

[13] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, PMLR, vol. 97, pp. 6105–6114, 2019.

[14] H. G. Ghifari, D. Darlis, and A. Hartaman, "Pendeteksi Golongan Darah Manusia Berbasis Tensorflow menggunakan ESP32-CAM," vol. 9, no. 2, pp. 359–373, 2021, doi: 10.26760/elkomika.v9i2.359.

[15] D. Kothadiya, N. Pise, and M. Bedekar, "Different Methods Review for Speech to Text and Text to Speech Conversion," *Int. J. Comput. Appl.*, vol. 175, no. 20, pp. 9–12, 2020, doi: 10.5120/ijca2020920727.