Bit-Tech (Binary Digital - Technology)
Vol.8, No.1, August 2025
Available online at: http://jurnal.kdi.or.id/index.php/bt

BISINDO Sign Language Interpreter System Using YOLOv8 and CNN

Farizah Farhana[1)*], Abdul Rezha Efrat Najaf[2)], Reisa Permatasari[3)]

[1)2)3)] *Information System, Computer Science Faculty, National Veteran Development University of East Java*

[1)]farhanafarizah@gmail.com

[2)]rezha.efrat.sifo@upnjatim.ac.id

[3)]reisa.permatasari.sifo@upnjatim.ac.id

Article history:

Received 11 June 2025;
Revised 15 June 2025;
Accepted 19 June 2025;
Available online 10 August 2025

Keywords:

BISINDO
CNN
Hand Movement Detection
Sign Language
YOLOv8

Abstract

Many deaf people in Indonesia use sign language as their communication medium, one of which is BISINDO (Bahasa Isyarat Indonesia). However, limited understanding of BISINDO among the general public often becomes a significant barrier to effective communication. To address this, the study aims to develop a BISINDO hand gesture detection system based on computer vision using the YOLOv8 algorithm. The method used is a Convolutional Neural Network (CNN) with the YOLOv8 architecture, trained on a labeled dataset obtained from the Roboflow platform consisting of BISINDO alphabet gestures (A–Z). The system was developed using Python and Streamlit, providing two types of user input: real-time camera feed and manual image upload. Detected gestures are translated into text and converted into speech using the Google Text-to-Speech (gTTS) API. The model was trained over 50 epochs and evaluated using metrics including accuracy, precision, recall, and F1-score. The evaluation results show an accuracy of 89.74%, precision of 89.28%, recall of 96.15%, and F1-score of 92.16%, indicating strong model performance and generalization. Some misclassifications occurred, such as 'R' detected as 'L', and background images mistakenly classified as valid letters. Nonetheless, the system is able to detect and translate BISINDO sign language gestures in semi-realtime with high reliability. This study contributes both practically and theoretically to the field of assistive technologies by providing an accessible, web-based platform for BISINDO recognition, promoting more inclusive communication for the Indonesian deaf community.

I. INTRODUCTION

Maulidia stated that sign language is the main tool used by deaf and mute people to communicate with other people [1]. In sign language, communication uses body language and lip movements as the main media to convey the meaning of a letter or word [2]. Hand gestures in sign language play an important role in conveying information, not just replacing voice [3]. Sign language in Indonesia is different from sign language in other countries [4], likewise, in Indonesia, there are two types, namely the Indonesian *Sistem Isyarat Bahasa Indonesia* (SIBI) and *Bahasa Isyarat Indonesia* (BISINDO). The main difference between SIBI and BISINDO is in the language structure and its use. SIBI is more formal because it follows the rules of Indonesian grammar. In contrast, BISINDO is more commonly used in everyday conversation. BISINDO itself grew naturally in the Indonesian deaf community even before independence [5].

Deaf friends often face various challenges in their daily lives, such as being left behind in getting information, having difficulty understanding lessons at school, and obstacles in communicating directly with doctors, police and other public services [5]. In many case, these barriers can reduce their independence and participation in society. The limited number of professional interpreters results in communication difficulties in various situations, this is because not everyone is able to use sign language [6]. This communication gap often makes deaf and mute people rely on writing or gestures that are not always clearly understood. Therefore, there is a need for technological assistance to help translate sign language into text or voice automatically.

Although sign language recognition systems exist, many are device-dependent or require installation. Most only recognize a limited range of gestures and lack flexibility. Web-based solutions for BISINDO remain scarce and underdeveloped. There is a clear need for a practical, accessible system that supports full BISINDO interpretation.

* Corresponding author

doi: 10.32877/bt.v8i1.2638

Rapidly developing technological advances, especially in the field of Computer Vision, help in automating various visual processes, including sign language translators. Various models in deep learning have shown significant success in image classification and object detection, which opens up the possibility of recognizing hand gestures more accurately and quickly. For now, the existence of a semi-realtime sign language translation detection system, which is able to translate hand gestures in sign language directly, will be very helpful in the interaction of deaf people with the wider community. This technology not only helps in communication, but also promotes inclusivity and equality in accessing services.

The most frequently used method in visual pattern detection is the Convolutional Neural Network (CNN) with th e You Only Look Once Version 8 (YOLOv8) architecture, the combination of both methods in pattern recognition will help make this technology very potential in automating sign language translation semi-realtime [7]. YOLOv8, as an advanced version of YOLO, has better performance in terms of accuracy and speed, making it suitable for real-time applications such as sign language recognition. CNN, on the other hand, is well known for its ability to learn spatial hierarchies from input images, which is essential in gesture recognition.

Most existing research has not fully explored the capabilities of YOLOv8 for BISINDO gesture recognition. Current systems are rarely optimized for real-time use in browser environments. In many cases, studies focus only on numbers or a few letters, not the complete A–Z BISINDO alphabet. This shows a gap in both technical approach and scope of gesture coverage.

Based on previous studies, there has not been much development of a BISINDO sign language translator that is specifically based on a website. Most of the existing systems are still sandalone applications that require installation or specific devices. With this technology, sign language can be directly translated into text and voice, which allows communication to be smoother, both in public services, education, and daily interactions. In addition, it is hoped that this system will be able to provide easy access for the general public without requiring the installation of special hardware or software, so that it can be widely used in various fields such as public services, education, and the world of work.

The goal is to improve equality of access and communication for people with disabilities who are deaf and mute in Indonesia. In addition, this study only focuses on BISINDO sign language, by detecting letters of the alphabet from A to Z BISINDO. This limitation is taken to ensure more accurate performance and to build the foundation for future research, which may expand into world-level or sentence-level recognition. This research addresses the gap by developing an accessible, semi-realtime BISINDO letter recognition system using YOLOv8, which can be accessed via a web platform without additional installations. This research is expected to contribute to efforts in building more inclusive and accessible digital environments in Indonesia.

II. LITERATURE REVIEW

Not many people understand about sign language, which is used as a means of communication for the deaf and mute. In communicating, sign language is done manually, using hand gestures, facial expressions, and body movements [8]. The Indonesian Sign Language Center (Pusbisindo) is an institution established to develop and research sign language in Indonesia, especially BISINDO. Pusbisindo is under Gerkatin and was established on February 28, 2009. Its main objective is to strengthen the status of sign language as a legitimate language and to provide support to the deaf community in Indonesia [9].

In this study, we used a deep learning approach, which is part of machine learning, to detect hand movements of the BISINDO alphabet letters through image input [10]. Deep learning utilizes deeper and more complex neural network structures to capture important features from images [11] [12]. One of the implementations is in object detection from convolutional neural networks (CNN) using the YOLOv8 architecture [13] which enables object identification both in digital images and in videos. [14]. You Only Look Once (YOLO) was first introduced in computer vision through a paper published by Joseph Redmon in 2015 titled "You Only Look Once: Unified, Real-Time Object Detection"[15].

Compared to earlier versions like YOLOv3 and YOLOv4, YOLOv8 offers improved detection speed, higher accuracy, and a more streamlined model structure, making it more suitable for real-time gesture recognition. CNN itself has been widely used in gesture classification due to its strength in learning spatial features, but when combined with YOLOv8, it provides both localization and classification in one unified model.

Previous studies on BISINDO sign language interpreter systems have generally utilized the CNN method in combination with the YOLO architecture, particularly YOLOv8, using datasets that consist of numbers, letters, and several words. Indah et al. applied CNN for hand gesture classification and YOLO for detecting hand gestures representing the numbers one to five, achieving an accuracy of 89% [2]. In a similar study, Lukman and Barka implemented CNN and YOLO for eight gesture classes—A, B, C, D, E, I, You, and I Love You—obtaining an accuracy of 99.82% after 1374 training epochs [5]. Alicia et al. utilized YOLOv4 to recognize the BISINDO alphabet using input images with a resolution of 1077×1077 pixels, achieving an accuracy of 99.4% [16]. In another study, Abdullah and Suci employed YOLOv3 for hand gesture detection and classification under varying background and lighting conditions, resulting in an accuracy above 90% [3]. Shamrat et al. conducted research on Bangla sign language using CNN to convert hand gesture patterns into text, reaching an accuracy of 99.8% [17].

However, most of these studies focused on limited gesture classes or required specific hardware and applications, making them less accessible to the general public. In addition, no prior research has combined YOLOv8 with a web-based deployment platform that includes both live detection and text-to-speech features for BISINDO. This highlights a practical and technical gap that this study aims to address.

Although various approaches have been developed, no study has specifically built a BISINDO translation system based on a website with both text and voice output. Therefore, this study proposes the use of YOLOv8 and gTTS in system development, which not only supports detection through a live camera but also provides an image upload feature as an alternative input method.

Google Text-to-Speech (gTTS) is a Python library that enables the conversion of written text into speech using Google's Text-to-Speech API. Instead of playing the audio directly, gTTS generates and saves the output as an MP3 file, which must then be played manually to hear the result. It supports a wide range of languages, including Indonesian. [18].

The BISINDO alphabet dataset used in this study was obtained from Roboflow, a web-based platform that provides access to various datasets. Roboflow also supports preprocessing, model training, and annotation using bounding boxes [19]. The model was trained using JupyterLab, a web-based interactive development environment that is part of Jupyter Notebook, supporting three programming languages: Julia, Python, and R [20]. For gesture pattern detection, this study utilizes a deep learning framework, including the OpenCV library, which is used for real-time dynamic image processing [21]. The entire system was developed using Python, a high-level programming language commonly used in software development, data analysis, and artificial intelligence [22]. The trained model was then deployed using Streamlit to facilitate public access. Streamlit is an open-source and free Python library for developing web applications [23].

Before the system is implemented, it is important to conduct testing to determine whether the system functions properly according to requirements and accuracy expectations [24]. The testing in this study was carried out in two stages: direct testing by users to evaluate the performance of the YOLOv8 model [25], and functional testing to ensure that each feature operates according to its intended purpose [24].

III. METHODS

A. Research Methods

This study aims to develop a BISINDO sign language translation system that converts hand gestures representing the A-Z alphabet into text and speech in semi-realtime. The system is built using the YOLOv8 method, which is based on CNN, to ensure accurate and efficient detection. Figure 1 illustrates the overal workflow of this research process.

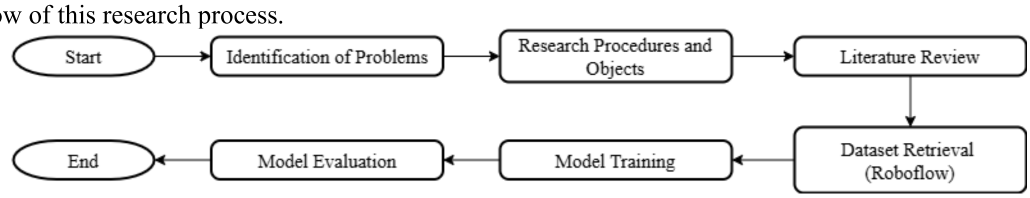


Fig. 1 Research flow process

The system begins with the collection of hand gesture data from the BISINDO alphabet, followed by training the YOLOv8 model using the dataset. Once the model is trained, its performance in detecting alphabet gestures is evaluated. The cllasification results are then converted into text and integrated with a Google Text-to-Speech (gTTS) model to generate voice output. This entire process is developend in the JupyterLab environment using Python, supported by the Ultralytics library. To ensure reproducibility, the entire process including dataset structure, model configuration, augmentation settings, and training logs was documented and version-controlled.

1. Research Stages

   The research begins with problem identification, followed by a literature review to explore realted studies and methods. Next, a dataset containing BISINDO alphabet hand gestures is collected and uset to train the YOLOv8 model. After the training process, the model is evaluated to measure its detection performance. Finally, the system is integrated to translate BISINDO alphabet gestures into text and speech. The stages are summarized in the flowchart shown in Figure 1.

2. Identification of Problem

   This study addresses the communication challenges faced by the deaf community, particularyly due to the lack of sign language interpreters in public spaces. Existing BISINDO translation systems are limited, often recognizing only a few letters or numbers and using outdated YOLO versions with lower accuracy and speed. Recognizing the full A-Z alphabet is essential for more flexible and inclusive communication. Therefore, this study uses the latest YOLOv8 model to develop a more complete and adaptive system that translates BISINDO gesture into text and speech in semi-realtime.

3. Research Procedures and Objects

   This study uses an experimental approach to evaluate the performance of the YOLOv8 model in recognizing BISINDO sign language letters based on detection accuracy and effectiveness. The focus lies in training and evaluating the model to translate BISINDO hand gestures into letters in semi-realtime. The dataset was obtained from the Roboflow platform, consisting of images of BISINDO alphabet gestures from A to Z, with 16-17 images per class. As an object of study, each gesture image was treated as an instance of single-class detection, where one bounding box and one label were used per image.

4. Dataset Retrieval

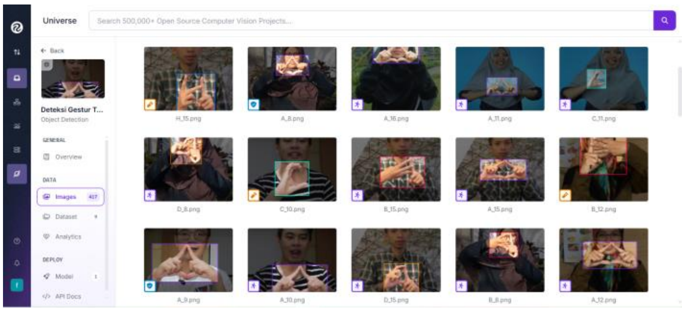


Fig. 2 Dataset

The data used in this research is for supervised learning and was obtained through the Roboflow platform [22], [23]. It consists of 26 letters of the BISINDO sign language alphabet, with each category containing 16 to 17 images. This dataset includes a total of 417 images, as shown in Figure 2.

The dataset obtained has undergone pre-processing in Roboflow, where it was organized based on its respective classes and labels, as shown in Figure 2. In addition, data variation was introduced through augmentation to make the model more robust and avoid overfitting. Specifically, 20% of the images were converted to grayscale, and random brightness adjustments were applied within a range of -25% to +25%. Each image was resized to 96 x 96 pixels.

The next stage involves splitting the annotated dataset into three parts: training set, validation set, and test set, with a proportion of 70-20-10 from the total of 417 images, as shown in Table 1 below:

TABLE 1
SPLITTING DATASET

Model	Proportion	Number of Images
Train	70%	292 Images
Valid	20%	83 Images
Test	10%	42 Images

All dataset processing and augmentation were performed using Roboflow's preprocessing pipeline. Bounding boxes were retained and adjusted accordingly to preserve detection consistency post-augmentation.

5. Model Training

The model was trained using the YOLOv8 architecture from Ultralytics with a BISINDO A–Z alphabet dataset obtained from Roboflow. Model training was carried out in JupyterLab using the dataset obtained from Roboflow. The training process involved importing the dataset configuration script provided by Roboflow, as shown in Figure 3. The data was trained using YOLOv8 with 50 epochs to produce the best model. The initial training configuration used an input resolution of 96×96 pixels. The dataset was automatically split into training (70%), validation (20%), and testing (10%) sets. Data augmentation was applied by converting 20% of the images to grayscale and randomly adjusting brightness levels between -25% and +25%.

The model was trained with a batch size of 16, learning rate of 0.01, and image size 96×96, using the SGD optimizer. Loss metrics and mAP (mean Average Precision) were tracked across epochs to monitor convergence. Training logs, loss curves, and validation results were saved as artifacts to enable reproducibility and performance comparison.

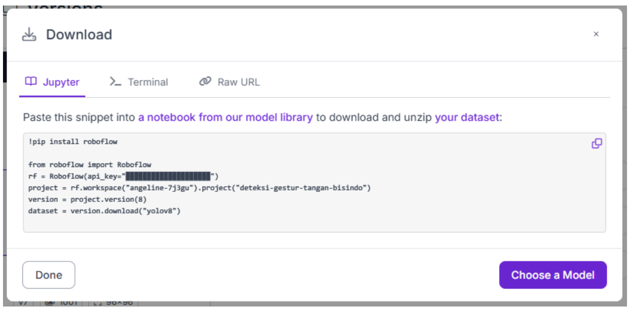


Fig. 3 Pelatihan Model

B. Implementation System

The implementation of this system aims to translate BISINDO alphabet letters in semi-realtime using the YOLOv8 model. The system supports two input methods: capturing images directly through a camera and uploading images from a device. The detection model used is the result of training with the BISINDO letter dataset, which was imported and trained in the JupyterLab environment using the Python programming language.

The trained YOLOv8 model is saved in .pt (PyTorch) format, allowing it to be used directly without retraining. When the system is run, users can choose the input method. If using a camera, the system will capture a hand gesture image in real time. If using the upload method, the system will process the image submitted through the interface. The image is then passed to the model for detection and classification into one of the BISINDO alphabet letters.

The detection results are displayed as text on the system interface. The main features implemented include detection of letters A to Z in BISINDO, support for both camera and image upload input methods, and automatic display of results. The detection process runs quickly, allowing the system to operate in semi-realtime. This implementation enables the system to run efficiently on a local machine and provides an interactive experience for users.

For evaluation, test images were passed through the trained model and predictions were compared against ground truth using confusion matrix analysis. Precision, recall, and F1-score metrics were also calculated to assess model performance.

IV. Results

A. Visualization of Training Results

The YOLOv8 model applied for detecting BISINDO hand shape letters showed positive progress during the training process. As illustrated in Figure 4, the box-loss, cls-loss, and dfl-loss graphs consistently decreased for both training and validation, indicating that the model successfully learned and optimized its parameters. In addition, the evaluation performance on metrics such as precision, recall, mAP50, and mAP50-95 also showed stable improvements, reflecting the model's increasing ability to accurately recognize objects.

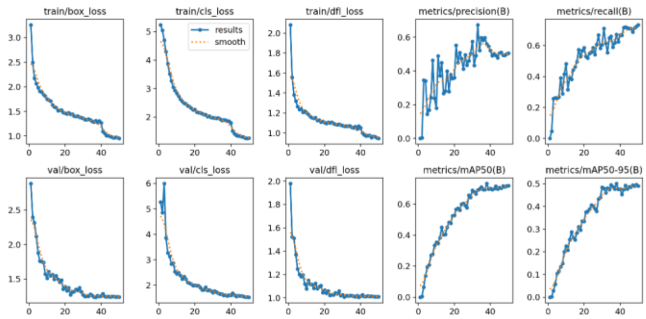


Fig. 4 Chart Loss Training

B. Results of System Detection Implementation

This detection system provides two input methods: capturing images directly using a camera and uploading images from the user's device. Users can choose either method via the sidebar menu available in the system interface. The following section presents the implementation results of each input method.

1. Take pictures directly

Fig. 5 Page taking pictures with camera

The input process using the camera requires the user to capture an image first by clicking the 'Take Photo' button, as shown in Figure 5. Once the image is captured, it will be processed directly by the system, and the recognized result will be displayed on the interface in the form of text output and accompanied by audio output to assist accessibility for users. This semi-realtime feedback helps users immediately know the detected BISINDO letter without needing additional input or navigation.

2. Upload Hand Image to System

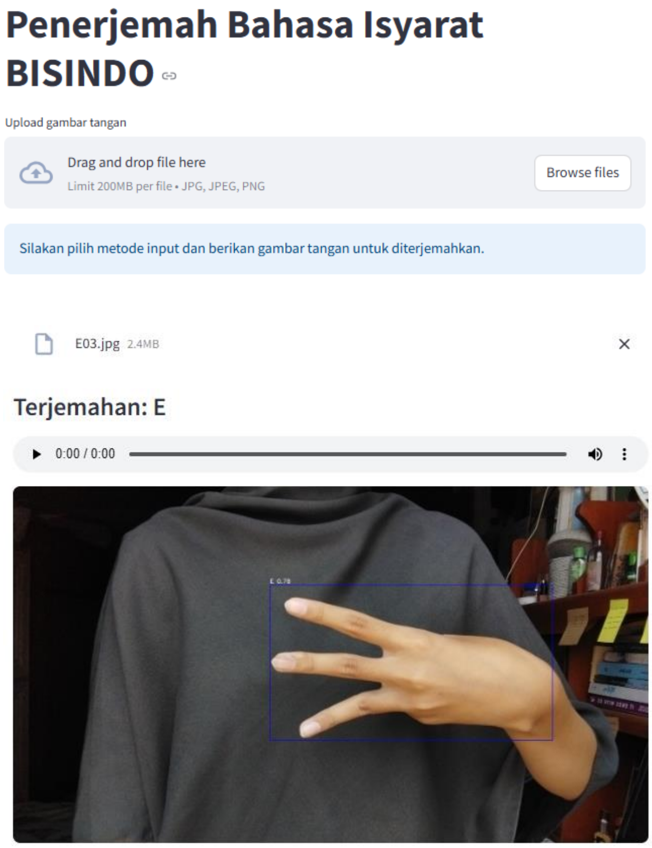


Fig. 6 Page Upload Image

Similar to the previous method, the image upload method requires the user to first upload an image to the system, as shown in Figure 6. The uploaded image will then be processed directly by the system, which will automatically display the detected text, audio output, and the confidence score of the result.

C. Model Performa Evaluation

1. Live testing on the system

TABLE 2
DIRECT TEST RESULT

Letter Input	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Image							
Letter Output	A	B	C	D	E	F	G
Audio Output	A	B	C	D	E	F	G
Detection Accuracy	0.56	0.97	0.68	0.67	0.96	0.30	0.86
Status	Valid	Valid	Valid	Valid	Valid	Valid	Valid
Letter Input	**H**	**I**	**J**	**K**	**L**	**M**	**N**
Image							
Letter Output	H	I	J	T, K	L	M	N
Audio Output	H	I	J	T, K	L	M	N
Detection Accuracy	0.91	0.46	0.89	0.55, 0.31	0.56	0.27	0.69
Status	Valid	Valid	Valid	Valid	Valid	Valid	Valid

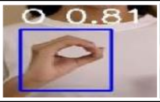
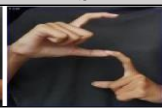
Letter Input	**O**	**P**	**Q**	**R**	**S**	**T**	**U**
Image							
Letter Output	O	T, P	Q	R	S	T	U
Audio Output	O	T, P	Q	R	S	T	U
Detection Accuracy	0.81	0.83, 0.44	0.25	0.51	0.50	0.82	0.26
Status	Valid	Valid	Valid	Valid	Valid	Valid	Valid
Letter Input	**V**	**W**	**X**	**Y**	**Z**	**-**	**-**
Image							
Letter Output	V	W	X	Y	Z	Detection results with dark lighting	Detection results with solid background
Audio Output	V	W	X	Y	Z		
Detection Accuracy	0.98	0.99	0.44	0.85	0.64		
Status	Valid	Valid	Valid	Valid	Valid		

Table 2 consists of six columns: image, input letter, text output, audio output, detection accuracy, and status. The image column displays the detection results with bounding boxes and confidence scores generated by YOLOv8, including variations in lighting conditions such as low-light and solid backgrounds to test the system's robustness. The detection accuracy shown in the table is based on the confidence score assigned by the YOLOv8 model for each recognized sign language letter in the input image. This score reflects how confident the model is that the detected object corresponds to the intended letter. The score ranges from 0 to 1, with a minimum threshold of 0.25. The status is marked "Valid" if the detection matches the expected input and "Invalid" if it does not. During testing, the letter R was incorrectly detected as L (False Negative), while letters K and P were detected even when no gesture was present (False Positive). The system also successfully ignored 10 images with no gestures (True Negative) and correctly identified 25 out of 26 letters (True Positive).

The overall accuracy of 89.74% was calculated using the confusion matrix formula: (TP + TN) / (TP + TN + FP + FN), with TP = 25, TN = 10, FP = 3, and FN = 1. This value demonstrates that the model is fairly reliable in consistently recognizing BISINDO alphabet gestures. Based on the results, The lowest confidence score was 0.25 on the letter Q, while the highest was 0.99 on the letter W. This range indicates that the majority of gestures were identified with high certainty, although some were predicted with lower confidence. The variation in confidence scores suggests that visually similar gestures or suboptimal lighting conditions may reduce the model's certainty in predictions. Therefore, improving data quality or applying further augmentation techniques could help enhance detection stability for specific gestures.

2. Functional testing on the system

TABLE 3
FUNCTIONAL TESTING RESULTS

Feature	Input	Expected Output	Results	Status
Take a picture with camera	Click "Take Photo" with the hand shape of the letters BISINDO	The image is shown in the UI and processed by the model.	The image appears and the system detects the letter with accuracy.	Valid
Upload an image of a BISINDO handshape letter.	Upload image letter E	The image is displayed in the UI, and the letter E is detected.	The letter E was successfully detected, with an accuracy	Valid
Terjemahan teks muncul The translated letter appears.	Hand gesture image is inputted.	Translated letter appears on the UI.	Translated letter label is displayed.	Valid
The audio is played	A different hand gesture image is input.	The system detects the label and plays the audio.	The audio is clearly heard.	Valid

The results of the functional testing show that all core features of the system operated effectively and aligned with the design objectives. Features such as capturing images via the camera, manually uploading images, translating hand gesture images into letters, playing the detected audio, and displaying detection accuracy on the image all functioned smoothly without issues. Evidence of this implementation is presented in Table 3, which summarizes the outcomes of each function.

3. Evaluation of model performance metrics

   YOLOv8 is a deep learning object detection model built upon Convolutional Neural Network (CNN) architecture. In this study, YOLOv8 is used to detect and classify hand gestures corresponding to BISINDO alphabet letters. The model processes input images by extracting spatial features through convolutional layers, generating bounding boxes around detected gestures, and assigning class labels with associated confidence scores. This end-to-end detection pipeline allows the system to recognize visual patterns in semi-realtime with high accuracy.

   The performance evaluation of the YOLOv8 model was carried out using metrics such as precision, recall, F1-Score, and mAP. The results show that the model achieved a precision of 89.28%, recall of 96.15%, and an F1-Score of 92.61% on the validation data. An accuracy of 89.74% was also recorded, indicating that the model is capable of detecting objects reasonably well. These findings support the model's reliability for use in a sign language translation system.

   Analysis of the confusion matrix revealed that some misclassifications occurred, such as the letter 'K' being detected as both 'K' and 'T', the letter 'P' being detected as both 'P' and 'T', and the letter 'R' being misclassified as 'L'. These errors likely result from similarities in hand shape and orientation, especially under varied lighting conditions.

   Furthermore, the model maintained consistent performance when tested with moderate changes in lighting and background, indicating a degree of robustness. However, minor false positives were observed where background images were mistakenly classified as letters, suggesting a sensitivity to image noise. These insights highlight both the strengths and current limitations of the model in real-world application.

V. Discussion

The implementation results of the BISINDO alphabet translation system show that the CNN method with the YOLOv8 architecture is capable of detecting A–Z hand gestures in a semi-realtime manner with good performance. The system was successfully run in a local environment using Python and a Streamlit interface, supporting input from both a live camera and image uploads. The detection results are displayed visually through bounding boxes and letter labels, and are also accompanied by audio output via gTTS integration.

The YOLOv8 model used was trained for 50 epochs with an annotated dataset from Roboflow. The dataset consists of BISINDO alphabet gesture images from A to Z, with 16–17 images per class. The training process was carried out in the JupyterLab environment and produced a model in .pt format, which can be used directly for detection without retraining. Image augmentation was applied to improve model generalization, including converting images to black and white and randomly adjusting brightness levels.

Model performance evaluation on the test data yielded a precision of 89.28%, recall of 96.15%, and an F1-score of 92.61%. The overall accuracy reached 89.74%, indicating the model's consistent ability to distinguish hand gestures. Based on the confusion matrix, there were 25 True Positives (TP), 10 True Negatives (TN), 3 False Positives (FP), and 1 False Negative (FN). Some detection errors were identified, such as the gesture R being detected as L (FN), and the letters K and P being detected even without any gesture input (FP).

These results demonstrate that the system successfully fulfills the main objective of this research: to develop a BISINDO alphabet translator into text and speech with high accuracy and practical accessibility. Moreover, the system addresses the research questions concerning the limitations of previous BISINDO translation systems, which only recognized a few letters or numbers and used older YOLO versions. By leveraging YOLOv8, this system provides improvements in detection speed and accuracy, making it a more adaptive and inclusive solution for early-stage BISINDO sign language translation.

Compared to previous studies, which used older YOLO versions (YOLOv3 or YOLOv4), the current system demonstrates competitive accuracy while offering additional functionality such as web-based access and voice output. This extends the practical usability of sign language recognition systems into more accessible domains. Analysis of the confusion matrix revealed that some misclassifications occurred, such as the letter 'K' being detected as both 'K' and 'T', the letter 'P' being detected as both 'P' and 'T', and the letter 'R' being misclassified as 'L'. The dataset size is relatively small (417 images total), which may limit the model's generalization in more diverse real-world conditions. The system also shows some sensitivity to lighting variation and may misclassify in low-light environments.

Future research should focus on expanding the dataset with more diverse and balanced samples, integrating real-time continuous gesture recognition (not just single letters), and improving robustness under uncontrolled lighting. Additionally, incorporating temporal sequence models such as LSTM or transformers could allow recognition of full words or phrases, enabling translation of complete sentences in BISINDO.

VI. CONCLUSION

This study successfully implemented a BISINDO alphabet translator system using the Convolutional Neural Network (CNN) method with the YOLOv8 architecture. The model was trained using an annotated dataset from Roboflow, consisting of BISINDO hand gesture images for letters A–Z. After training for 50 epochs in the JupyterLab environment, the model was integrated into a Python-based Streamlit application that supports both live camera and image upload input. The system outputs detected letters visually with bounding boxes and class labels, and also converts the results into audio using gTTS.

The evaluation results demonstrate that the system performs well in recognizing BISINDO hand gestures, achieving a precision of 89.28%, recall of 96.15%, F1-Score of 92.61%, and overall accuracy of 89.74%. The confusion matrix further supports the model's effectiveness, with 25 true positives, 10 true negatives, 3 false positives, and 1 false negative. These findings confirm that the system can accurately detect and translate BISINDO letters into text and speech, fulfilling the research objectives despite minor detection errors in some conditions.

Theoretically, this research contributes to the development of computer vision applications by demonstrating the effectiveness of YOLOv8 in sign language recognition, particularly within the underexplored context of BISINDO. Practically, the implementation of a web-based, semi-realtime translation system opens new possibilities for inclusive communication tools, accessible without the need for specialized hardware. By combining object detection with audio output, this system bridges gaps in accessibility, supporting deaf individuals in more diverse social settings such as education, healthcare, and public services. This study also lays the groundwork for future development of full-sentence BISINDO recognition systems, which could further advance automated sign language interpretation in Indonesia.

REFERENCES

[1] Rian Nanda, “Perancangan Aplikasi Tuna Wicara Dan Tuna Rungu Dengan Metode Waterfall Berbasis Android,” *JEKIN - J. Tek. Inform.*, vol. 3, no. 1, pp. 20–30, 2022, doi: 10.58794/jekin.v3i1.189.

[2] I. I. Arifah, F. N. Fajri, and G. Q. O. Pratamasunu, “Deteksi Tangan Otomatis Pada Video Percakapan Bahasa Isyarat Indonesia Menggunakan Metode YOLO Dan CNN,” *J. Appl. Informatics Comput.*, vol. 6, no. 2, pp. 171–176, 2022, doi: 10.30871/jaic.v6i2.4694.

[3] A. Sani and S. Rahmadinni, “Deteksi Gestur Tangan Berbasis Pengolahan Citra,” *J. Rekayasa Elektr.*, vol. 18, no. 2, pp. 115–124, 2022, doi: 10.17529/jre.v18i2.25147.

[4] H. Halim and L. Lina, “Aplikasi Pengidentifikasi Bahasa Isyarat Berdasarkan Gerak Tubuh Secara Real Time Menggunakan Yolo,” *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 8, no. 2, pp. 300–304, 2023, doi: 10.51876/simtek.v8i2.215.

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

[6] I. Sari, Fivrenodi, E. Altiarika, and Sarwindah, “Sistem Pengembangan Bahasa Isyarat Untuk Berkomunikasi dengan Penyandang Disabilitas (Tunarungu),” *J. Inf. Technol. Soc.*, vol. 1, no. 1, pp. 20–25, 2023, doi: 10.35438/jits.v1i1.21.

[7] Egga Naufal Daffa Tanadi, Dhian Satria Yudha Kartika, and Abdul Rezha Efrat Najaf, “Sistem Pendeteksi Penyakit Kanker Kulit Menggunakan Convolutional Neural Network Arsitektur YOLOv8 Berbasis Website,” *Repeater Publ. Tek. Inform. dan Jar.*, vol. 2, no. 3, pp. 166–177, 2024, doi: 10.62951/repeater.v2i3.124.

[8] A. Santoso and G. Ariyanto, “Implementation of Hard-Based Deep Learning for Face Recognition,” *Emit. J. Tek. Elektro*, vol. 18, no. 01, pp. 15–21, 2018.

[9] “PUSBISINDO: Pusat Bahasa Isyarat Indonesia.” [Online]. Available: https://www.pusbisindo.org/

[10] K. Azmi, S. Defit, and Sumijan, “Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat,” *J. Unitek*, vol. 16, no. 1, pp. 28–40, 2023, doi: 10.52072/unitek.v16i1.504.

[11] M. Sholawati, K. Auliasari, and F. Ariwibisono, “Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (Cnn),” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 6, no. 1, pp. 134–144, 2022, doi: 10.36040/jati.v6i1.4507.

[12] Sidharth, “Major Sub-Fields in Artificial Intelligence (AI) You Need to Know,” *PyCodeMates*, 2023. [Online]. Available: https://www.pycodemates.com/2023/07/major-sub-fields-in-artificial-intelligence.html

[13] B. Mahesh, “Machine Learning Algorithms - A Review,” *Int. J. Sci. Res.*, vol. 9, no. 1, pp. 381–386, 2020, doi: 10.21275/art20203995.

[14] G. N. Rizkatama, A. Nugroho, and A. F. Suni, “Sistem Cerdas Penghitung Jumlah Mobil untuk Mengetahui Ketersediaan Lahan Parkir berbasis Python dan YOLO v4,” *Edu Komputika J.*, vol. 8, no. 2, pp. 91–99, 2021, doi: 10.15294/edukomputika.v8i2.47865.

[15] M. Hussain, “YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital

Manufacturing and Industrial Defect Detection," *Machines*, vol. 11, no. 7, 2023, doi: 10.3390/machines11070677.

[16] A. Kinanti, M. A. Afandi, I. Permatasari, and N. Y. Tarigan, "Deteksi Objek Bahasa Isyarat Alfabet BISINDO Menggunakan Deep Learning dan Arsitektur YOLO," *Techno.Com*, vol. 23, no. 2, pp. 411–419, 2024, doi: 10.62411/tc.v23i2.9889.

[17] F. M. J. M. Shamrat, S. Chakraborty, M. M. Billah, M. Kabir, N. S. Shadin, and S. Sanjana, "Bangla numerical sign language recognition using convolutional neural networks," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 23, no. 1, pp. 405–413, 2021, doi: 10.11591/ijeecs.v23.i1.pp405-413.

[18] J. D. Radana, "Implementasi Deep Learning Menggunakan Algoritma Convolutional Neural Network (CNN) untuk Image Classification Varietas Tanaman Tembakau," *Jawa Timur Fak. Teknol. Pertan. Univ. JEMBER*, 2022.

[19] A. Agustinus, R. Kurniawan, and H. Oktafia Lingga Wijaya, "Klasifikasi Emosi Melalui Ekspresi Wajah Menggunakan Algoritma Deep Learning," *Econ. Soc. Sci. Comput. Agric. Fish. 2nd 2023*, pp. 1215–1221, 2023, [Online]. Available: https://semnas.univbinainsan.ac.id/index.php/escaf/article/view/470

[20] A. Zein, "Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 28, no. 2, pp. 22–26, 2018, doi: 10.37277/stch.v28i2.238.

[21] Y. P. Iswoyo, R. Wulanningrum, and A. Bagus, "Identifikasi Jenis Burung Menggunakan Yolo8 Berbasis Web Streamlit," vol. 8, pp. 8–15, 2024.

[22] N. J. Hayati, D. Singasatia, and M. R. Muttaqin, "Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan," *Komputa J. Ilm. Komput. dan Inform.*, vol. 12, no. 2, pp. 91–99, 2023, doi: 10.34010/komputa.v12i2.10654.

[23] A. Ritonga and Y. Yahfizham, "Studi Literatur Perbandingan Bahasa Pemrograman C++ dan Bahasa Pemrograman Python pada Algoritma Pemrograman," *J. Tek. Inform. dan Teknol. Inf.*, vol. 3, no. 3, pp. 56–63, 2023, doi: 10.55606/jutiti.v3i3.2863.

[24] F. Damatraseta, R. Novariany, and M. A. Ridhani, "Real-time BISINDO Hand Gesture Detection and Recognition with Deep Learning CNN," *J. Inform. Kesatuan*, vol. 1, no. 1, pp. 71–80, 2021, doi: 10.37641/jikes.v1i1.774.

[25] Ainurrohma, "Akurasi Algoritma Klasifikasi pada Software Rapidminer dan Weka," *Prism. Pros. Semin. Nas. Mat.*, vol. 4, pp. 493–499, 2021, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/prisma/