

SIBI-Based Gesture Recognition System Using Random Forest for Hearing-Impaired Communication

Andre Pratama^{1*}, Ahmad Jurnaidi Wahidin²

^{1,2}Information Technology Study Program, Faculty of Engineering and Informatics, Bina Sarana Informatika University
Jl. Kramat Raya No.98, RT.2/RW.9, Kwitang, Senen District, Central Jakarta City, Special Capital Region of Jakarta 10450, Indonesia

e-mail: ¹fourtamatech@gmail.com, ²ahmad.ajn@bsi.ac.id

(*) Corresponding Author

Article Info: Received: 19-10-2025 | Revised : 23-12-2025 | Accepted : 09-01-2026

Individuals with hearing impairments often face communication barriers when interacting with people unfamiliar with sign language. One officially recognized sign system in Indonesia is the Indonesian Sign System (SIBI), which conveys meaning through hand gestures. Most existing sign language recognition studies focus on single-hand gestures, limiting expressiveness. This study proposes a two-hand gesture recognition system based on digital image processing to translate SIBI gestures into alphabetic letters, while additional gestures enable text control functions. The dataset consists of 29 gesture classes with 1,000 images per class, totaling 29,000 images, and is divided into training and testing sets using a train–test split. A Random Forest classifier is employed to handle high-dimensional landmark coordinate data. Experimental results demonstrate a classification accuracy of 99.97%. The system is implemented as a real-time, user-friendly application. Although high accuracy is achieved, potential overfitting due to the controlled dataset is identified as a limitation. Future work will focus on improving generalization using more diverse real-world data.

Keywords : Hearing Impairment, Sign Language, SIBI, Gesture Recognition, Random Forest, Machine Learning

INTRODUCTION

For individuals with hearing impairments, communication takes place in a silent but expressive world. Their thoughts and feelings are conveyed through body language and hand movements, which serve as their primary voice. Unfortunately, most people cannot understand this form of communication, creating significant barriers in daily interaction. In Indonesia alone, there are 27,983 Deaf students across public and private schools (Aptik, 2023), showing the urgent need for more inclusive communication tools.

Deaf individuals are those who cannot hear sounds due to physical limitations, making it difficult for them to communicate using speech (Hidayat, 2024). They communicate through facial expressions, natural gestures, and sign languages such as BISINDO or the Indonesian Sign System (SIBI). SIBI follows Indonesian grammar and uses one hand to represent letters (Sari, 2021), while BISINDO is more natural and develops through daily interactions within the Deaf community (Citra, 2024). These differences illustrate how communication challenges arise, especially when many people cannot “hear” sign language (Febriansyah, 2022).

Advancements in Artificial Intelligence and the digital revolution have transformed computers into systems capable of recognizing and learning patterns through Computer Vision (Marpaung et al., 2022). This technology has broad applications ranging from healthcare and surveillance to robotics and autonomous systems—and is increasingly relevant in supporting human communication.

By combining Computer Vision, Machine Learning (Permana et al., 2023), and image processing techniques, it is now possible to build a system that reads SIBI hand gestures and translates them into letters or meaningful words. This study develops a real-time gesture translation system using OpenCV (Srimulia, 2022), MediaPipe (Kukul & Durai, 2022), NumPy (Radya, 2024), and the Random Forest algorithm. The system can detect gestures, identify hand positions, and classify letters with a high accuracy of 99.97%.

Most previous gesture recognition studies focus on translating single-hand gestures into text, with limited interactive features such as text editing, spacing, deletion, or speech output, which restricts practical communication use. In contrast, this study introduces a dual-hand interaction approach, where one hand recognizes SIBI gestures and the other controls commands such as spacing, deletion, and text-to-speech, enabling more natural and efficient communication.



Based on this background, the aim of this research is to design a SIBI-based gesture translation program called Gesture Recognition (Oudah et al., 2020), which supports communication between individuals with hearing impairments and the general public (Nugroho et al., 2021). Practically, this application is expected to reduce communication barriers in daily interactions by integrating gesture recognition and interactive control features within a single system, and serve as an educational tool. Academically, the study contributes to the development of knowledge in Computer Vision and Artificial Intelligence (Hindarto et al., 2023), particularly in the design of interactive, real-time gesture recognition systems.

RESEARCH METHOD

This study collected data through observations of communication activities among individuals with hearing impairments, interviews with special education teachers, and literature reviews on sign language, computer vision, and machine learning. The software was developed using a prototyping approach involving iterative model development, user testing, and refinement based on user feedback.

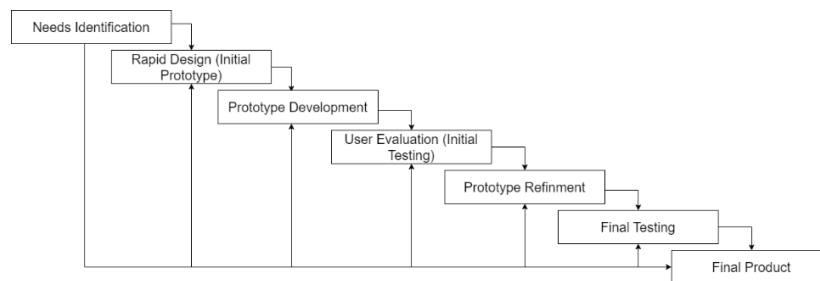
In its training process, this machine learning system employs the Random Forest method (Abdi et al., 2024). This algorithm was chosen for its high accuracy and robustness in handling complex data. Random Forest effectively manages large datasets and reduces overfitting by combining multiple decision trees, making it suitable for recognizing SIBI gesture letters in this study. Several SIBI gestures used in the research are shown in Figure 1.



Source: (Prasanda, 2022)

Figure 1. SIBI gesture images

In the application development process using the prototype method, the following figure illustrates the stages of the method. This approach allows continuous interaction between users and developers until the final system meets the desired requirements. The workflow of the system is shown in Figure 2.



Source: Research Results (2025)

Figure 2. Prototyping-Based Application Design Method Workflow

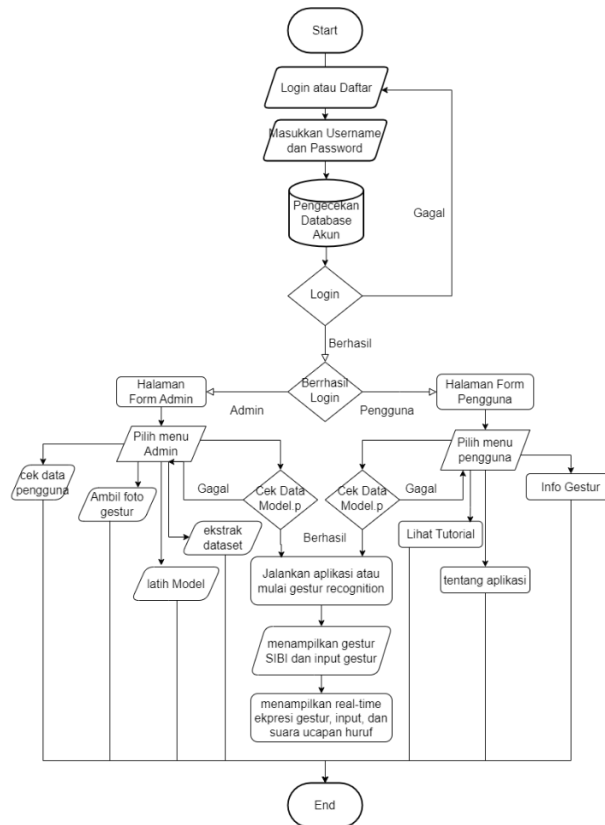
Based on Figure 2, this study applied the Prototyping Model, which was suitable because the system required repeated testing with real users—teachers and Deaf students—to ensure that gesture recognition matched actual communication needs. This method allowed early system versions to be evaluated and refined through continuous feedback.

The development process began with needs identification, including observing Deaf students’ communication, interviewing special education teachers, and reviewing literature on SIBI, Computer Vision, and Machine Learning.

A rapid design phase followed, producing the system’s basic structure and UML diagrams (Use Case, Sequence, Deployment, Flowchart, and ERD). An initial prototype was also created with features such as login, gesture input, gesture checking, and a feature extractor.

The prototype was then developed using OpenCV, MediaPipe, NumPy, and the Random Forest algorithm. A dataset of SIBI gesture images and videos was collected, and all features such as “Start Gesture Recognition,” “Check Gesture,” and “View Tutorial” were integrated.

A flowchart is used to describe the overall workflow of the system. It illustrates the sequence of processes, user interactions, and decision points within the application. Figure 3 shows the flowchart of the SIBI Gesture Recognition Application.



Source: Research Results (2025)

Figure 3. Prototyping-Based Application Design Method Workflow

Based on the flowchart in Figure 3, the application process begins with login or registration and account verification. Verified users then access the main menu according to their roles as administrators or users. The flowchart also illustrates the gesture data processing stages, including dataset extraction, model training, and gesture recognition. Finally, the system displays real-time recognition results, including gesture expressions, input visualization, and corresponding spoken letters. This workflow reflects the prototyping-based development approach used in this study.

In the initial development stage, a dataset of hand gesture images was collected using an external camera (a smartphone connected via USB). Each gesture class consisted of 1,000 images, resulting in a total of 29,000 images across 29 SIBi gesture classes. Sample images are shown in Figure 4.



Source: Research Results (2025)

Figure 4. Collection of Gesture Image Dataset

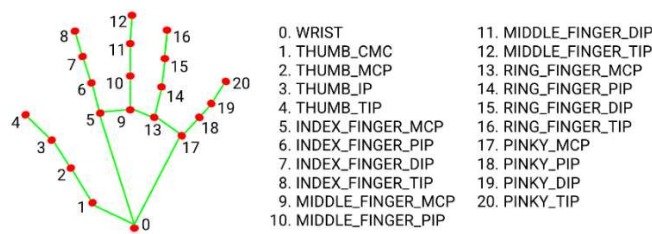
As shown in Figures 4, the dataset must have a balanced distribution to avoid bias during the model training process. Subsequently, the image data were processed using MediaPipe to extract the hand landmark coordinates, and the extracted dataset was compiled into a single data file in the form of a pickle. The illustration is presented in Figure 5.



Source: Research Results (2025)

Figure 5. Illustration of Landmark Conversion into Coordinate Points

In Figure 5, the landmark data obtained are then used as input features for training with the Random Forest method. At the implementation stage, the detection process is carried out in real time, enabling the system to directly recognize hand movement patterns during use. The figure 6 below illustrates the hand landmarks, including their numbers, associated regions, and designated names.



Source: (MediaPipe, 2023)

Figure 6. Figure showing the landmarks with their numbers and names

Based on the finger patterns in Figure 6, 21 landmark points—including fingertips, knuckles, and the wrist—are identified as numerical features that represent specific hand shapes or movements. These features are then learned by the model to consistently recognize letters or gestures.

The process involves several simple steps. First, the (x, y) coordinates of each hand point are extracted, typically ranging from 0 to 1 within the image frame. Then, the coordinates are normalized by subtracting the minimum x and y values from all points to ensure that recognition is not affected by the hand's position in the image.

$$x_{\min} = \min(x_1, x_2, \dots, x_{21})$$

$$y_{\min} = \min(y_1, y_2, \dots, y_{21})$$

Then, for each point, the x and y values are subtracted by these minimum values.

$$x'_i = x_i - x_{\min}$$

$$y'_i = y_i - y_{\min}$$

The purpose of this process is to shift all coordinates so that they are positioned relative to the top-leftmost point of the hand. After normalization, all coordinate values are arranged into a single row of data, forming what is known as a feature vector, as illustrated in the following example.

$$\text{fitur} = [x'_1, y'_1, x'_2, y'_2, \dots, x'_{21}, y'_{21}]$$

The processing generates a feature vector that numerically represents the hand-shape pattern. This vector becomes the input to the Decision Tree model during training and recognition, making landmark extraction an essential bridge between image data and classification.

In the training stage, the Random Forest algorithm builds multiple Decision Trees. Each tree splits the data using metrics such as Gini Impurity or entropy. Gini Impurity measures how mixed the classes are in a node—the lower the value, the purer the node. A Gini value of zero means the node contains only one class.

$$G = 1 - \sum_{i=1}^n p_i^2$$

p_i represents the proportion of data belonging to the i -th class (for example, class A, B, or C), while n denotes the total number of classes. Meanwhile, Entropy (Information Gain) measures the level of disorder (impurity) within a node. The higher the entropy value, the more random or mixed the data is. This metric helps determine how effectively a feature can split the data into distinct classes, aiming to reduce uncertainty and increase the purity of the resulting nodes.

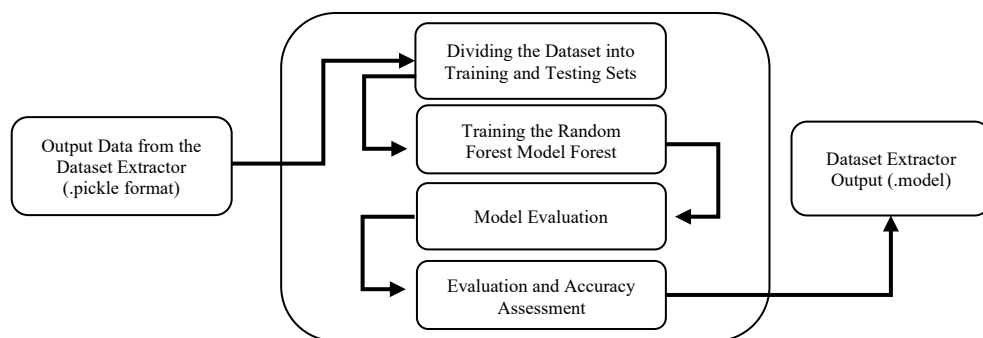
$$Entropy = - \sum_{i=1}^n p_i \cdot \log_2(p_i)$$

p_i represents the proportion of data belonging to the i -th class, and n denotes the total number of classes. The purpose of this splitting process is to find the optimal division that separates the data into the purest (most homogeneous) classes. Finally, the overall prediction is determined through a majority voting process among all the trees in the forest.

The extracted features were first saved in a .pickle file to efficiently store and reuse the feature vectors without repeated preprocessing. These stored features were then used to train the Random Forest model, ensuring consistent input during training. After training, the finalized model was saved as a .model file, allowing it to be reused for classification without retraining. This workflow improved efficiency and ensured reproducibility from data extraction to model deployment.

The process begins with the feature vectors and labels generated by the Dataset Extractor and stored in the .pickle file. These features are split into training and testing sets. During training, the Random Forest algorithm builds multiple decision trees using random subsets of the data and features, increasing model robustness and reducing overfitting. Once trained, the model is saved for future gesture recognition tasks.

The testing set is then used to evaluate the model's performance on unseen data. Metrics such as accuracy, precision, and recall are measured to validate the model's effectiveness in recognizing different hand gesture patterns. The overall Random Forest workflow is illustrated in Figure 7.



Source: Research Results (2025)

Figure 7. The training process of the Random Forest

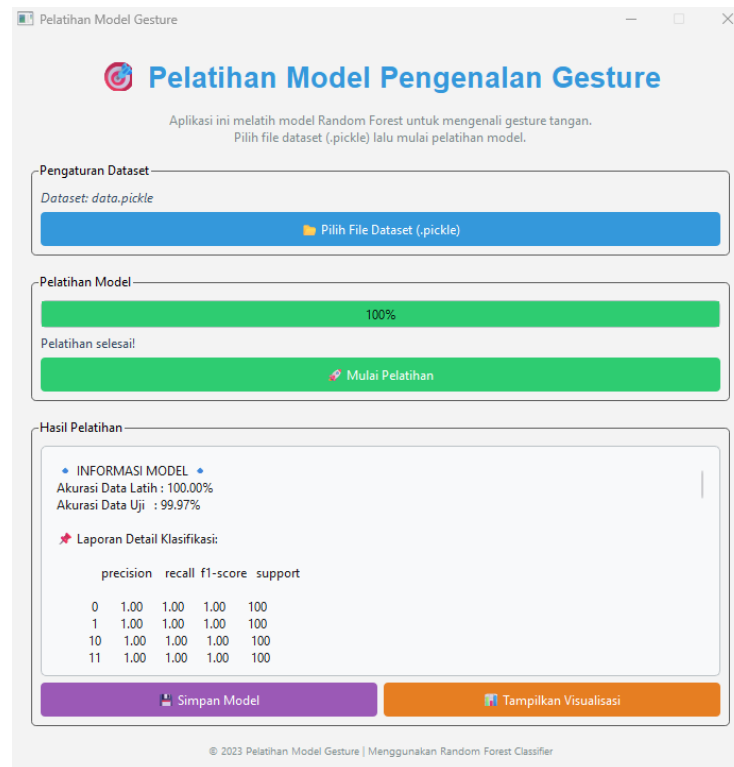
Random Forest was chosen as the classification algorithm because machine learning models cannot directly process raw image data. The images were therefore converted into simplified hand-landmark coordinates, which serve as numerical features that the algorithm can analyze. Using these features, Random Forest builds the gesture classification model.

The algorithm also offers strong performance on large and complex datasets and helps reduce overfitting through its ensemble of multiple decision trees. This makes Random Forest highly suitable for image-based gesture recognition.

RESULTS AND DISCUSSION

This section presents the results of system implementation and evaluation of the SIBI-based gesture recognition application. The discussion focuses on the training and testing outcomes, system performance, and user evaluation obtained after the development process was completed.

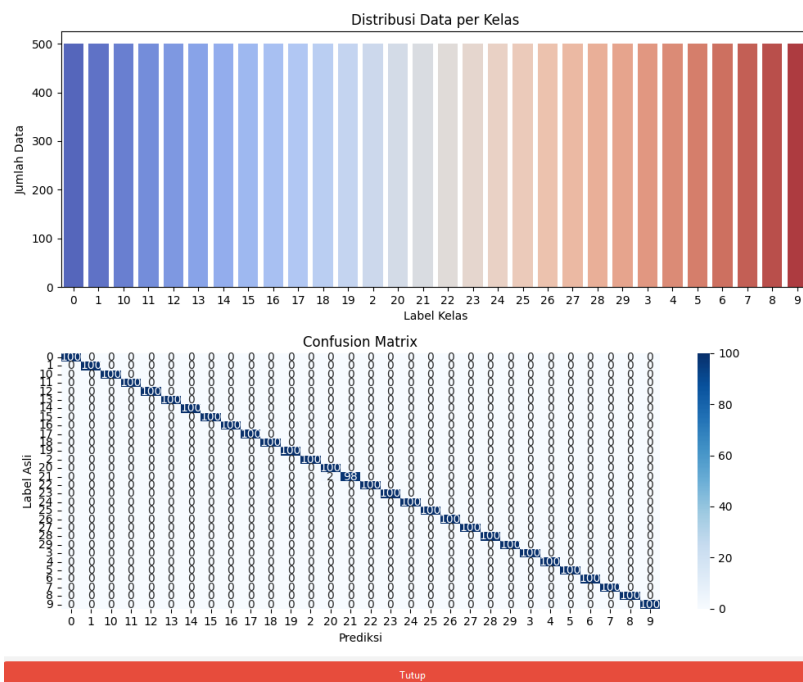
System training was conducted through the Gesture Model Training menu. The training data came from gesture images that were converted into landmark coordinates using MediaPipe. The design of this training process is shown in Figure 8.



Source: Research Results (2025)

Figure 8. Information on Gesture Model Training In The Application

In Figure 8, the training process was conducted using a supervised learning approach to develop the gesture recognition model (Wijoyo A et al., 2024). The dataset was divided into 80% for training and 20% for testing. The training results of all gesture accuracies are presented in Figure 8, where the model demonstrated stable and consistent convergence. The evaluation of the trained model showed that the system achieved an accuracy of 99.97% on the test data and 100.00% on the training data. These results indicate that the Random Forest algorithm is capable of classifying SIBI gesture letters with almost no errors. The accuracy values after training are shown in Figure 9.



Source: Research Results (2025)

Figure 9. Training Accuracy Values Using Random Forest In The Application

In Figure 9, several gesture training accuracy results are presented with the following descriptions:

1. Distribution of Data by Class

The data distribution graph shows that the number of samples in each class (0–29) is relatively balanced (around 500 samples per class). This ensures that the model training process does not suffer from bias caused by class imbalance. With evenly distributed data, Random Forest can learn the patterns of each class fairly.

2. Confusion Matrix

The evaluation results using the confusion matrix show that all predictions lie exactly on the main diagonal with a value of 100%. This indicates that the Random Forest model is able to classify the test data correctly in every class without any prediction errors.

The accuracy of this application was then implemented to evaluate its real-world performance. At the implementation stage, the system was tested to recognize letters A to Z with various hand movements. The testing results showed that the system was able to recognize the letters with high accuracy, even when performed by different users.

The menus in this application still consist of commonly used components such as image or dataset acquisition, dataset extraction, model training, and application testing. The interface of the gesture detection application is shown in Figure 10.



Source: Research Results (2025)

Figure 10. Interface of the Gesture Recognition Page In The Application

In Figure 10, the application interface was designed to be simple in order to ensure ease of use for the users. The main display consists of right and left camera areas to capture hand movements, with the following differences:

1. Main Camera (Right Camera), used to capture gestures A–Z and other gestures, as well as to display SIBI letter gestures.
2. Secondary Camera (Left Camera), used to capture gestures of the left hand while also serving as the input for gesture letters displayed on the right hand.

The prediction results function to display letters according to the recognized gestures, Check Gesture is used to view the SIBI gestures detected or utilized by the application, and History Gesture shows the history of gestures performed. This interface was developed using the PyQt6 framework in Python, with OpenCV integration for image processing.

The application is designed to be user-friendly and does not require advanced technical skills. It provides a structured interaction workflow in which the right hand is used to perform SIBI letter gestures, while the left hand functions as a control mechanism for adding characters, deleting input, and inserting spaces.

Users present letter gestures in front of the camera for recognition, and confirmed letters are added through control gestures. Space insertion and text-to-speech activation are also performed using predefined left-hand gestures, allowing the system to read the formed words aloud.

The accuracy of the Random Forest–based application was evaluated using a black-box testing approach, which assesses system outputs based on given inputs without examining internal processes. Model performance was measured by the accuracy of the resulting predictions. The black-box testing results are presented in the following table.

Table 1. Black Box Testing

No	Test Scenario	Test Case	Expected Results	Experimental Results	Conclusion
1	The testing was carried out on letters A–Z using right-hand finger gestures while the gesture recognition camera was operating.	Displaying hand gestures representing the alphabet letters A–Z based on the SIBI standard.	The real-time detection system successfully recognizes SIBI gestures corresponding to the alphabet letters A–Z, accurately matching the expected hand shapes and displaying the recognized letters on the main interface.	The system misclassified only the gesture representing the letter O as C, whereas all other gestures were accurately identified.	Almost Valid
2	The space, input, or delete gesture was tested using the left hand while the gesture recognition system was actively running.	Displaying a dedicated hand gesture representing the space, input, or delete command.	The dedicated space gesture is identified as a space input and triggers the text-to-voice function to vocalize the generated word, input to register letters detected, or delete to erase one character from the constructed word.	Meets Expectations	Valid
3	Executing the photo data acquisition application.	Acquiring image data according to the defined number of samples and gesture classes.	The system successfully acquires and saves image data based on the configured frame count and selected gesture classes.	Meets Expectations	Valid
4	Executing the image data extraction process.	Extracting image data into a matrix representation of finger landmark coordinates.	The system processes and transforms the image data into a structured dataset.	Meets Expectations	Valid
5	Executing the Random Forest model training process.	Conducting model training on the extracted dataset.	Presenting the training outcomes, including the generated model data and corresponding accuracy results.	Meets Expectations	Valid

Source: Research Results (2025)

Overall, the developed SIBI-based gesture recognition system successfully meets the objectives of this study, which aim to provide an application that assists individuals with hearing impairments in communication. The high recognition accuracy and the simple interface design indicate that the application has strong potential to be used as an alternative communication medium in real-world situations.

To support this result, the prototype was tested with teachers and Deaf students after the core functions had been implemented. Feedback from these users focused on usability and recognition accuracy and was used to refine both the classification model and the user interface. One of the main improvements addressed the frequent misclassification of the letter “C” as “O.” After these improvements, final evaluation using Black Box Testing confirmed that all features worked properly. The Random Forest model reached 99.97% accuracy, demonstrating that the application can operate reliably in real time and effectively support communication between Deaf individuals and the general public (Febriyanti et al., 2021).

To position this study within existing research, a comparison was made with the previous work titled “Classification of SIBI Alphabet (Indonesian Sign Language System) Using MediaPipe with Deep Learning Method”. In that study, the authors implemented a Fully Connected Layer (FCL) deep learning model and achieved 94.32% accuracy on training data (Maryamah et al., 2023). In contrast, this study employs the Random Forest algorithm, which demonstrates stable performance and offers a simpler alternative to deep learning, particularly for real-time systems with low computational requirements.

CONCLUSIONS

This study demonstrates that the developed SIBI-based hand gesture recognition application performs effectively in real-time conditions. Using MediaPipe and OpenCV for hand landmark detection and a Random Forest classifier for gesture recognition, the system achieves an accuracy of 99.97%, indicating reliable performance.

The proposed system supports communication for individuals with hearing impairments by translating SIBI hand gestures into understandable outputs for non-disabled users. Most SIBI alphabet gestures are recognized accurately, although minor misclassifications occur for visually similar gestures such as “O” and “C”.

Despite its high accuracy, the study has limitations, including gesture similarity and the current desktop-based implementation. Future work will focus on expanding the dataset, developing mobile-based platforms, and exploring dual-hand recognition and more advanced AI methods to enhance usability and performance.

REFERENCES

- Abdi, K., Warjaya, A., Muthmainnah, I., & Pahutar, P. H. (2024). Penerapan Algoritma Random Forest dalam Prediksi Kelayakan Air Minum. *Jurnal Ilmu Komputer Dan Informatika*, 3(2), 81–88. <https://doi.org/10.54082/jiki.81>
- Aptik, L. (2023). *Pendidikan Anak Tunarungu (Tuli) di Indonesia: Apakah Sudah Memadai?* Kompasiana. <https://www.kompasiana.com/laurensiaaptik7925/652e051dec794a5900737232/pendidikan-anak-tunarung-tuli-di-indonesia-apa-ah-sudah-memadai>
- Citra, E. (2024). *Perbedaan Bahasa Isyarat: BISINDO VS SIBI*. Lembaga Bahasa Internasional Fakultas Ilmu Pengetahuan Budaya Universitas Indonesia (LBI FIB UI). <https://lbifib.ui.ac.id/id/blog/artikel/perbedaan-bahasa-isyarat-bisindo-vs-sibi>
- Febriansyah, F. D. (2022). *Bahasa Isyarat : Pengertian, Sejarah, dan Jenis-jenisnya [Lengkap]*. Ayo Berbahasa. <https://www.ayo-berbahasa.id/2022/09/bahasa-isyarat.html>
- Febriyanti, N. M. D., Sudana, A., & ... (2021). Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen. *Jurnal Ilmiah* ..., 2(3). [http://download.garuda.kemdikbud.go.id/article.php?article=3457876&val=30165&title=Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen](http://download.garuda.kemdikbud.go.id/article.php?article=3457876&val=30165&title=Implementasi%20Black%20Box%20Testing%20pada%20Sistem%20Informasi%20Manajemen%20Dosen)
- Hidayat, I. (2024). *Sama-Sama Tuna, Apa Bedanya Tuna Rungu dan Tuna Wicara?* TambahPinter.Com. <https://tambahpinter.com/perbedaan-tuna-rungu-dan-tuna-wicara/>
- Hindarto, H., Sumarno, S., & Rosid, M. A. (2023). Buku Ajar Kecerdasan Buatan/Artificial Intelegent (AI). In *Buku Ajar Kecerdasan Buatan/Artificial Intelegent (AI)* (Issue September 2022). <https://doi.org/10.21070/2022/978-623-464-034-2>
- Kukil & Durai. (2022). *MediaPipe – The Ultimate Guide to Video Processing*. LearnOpenCV. <https://learnopencv.com/introduction-to-mediapipe/#What-is-MediaPipe?>
- Marpaung, F., Aulia, F., & Nabila, R. C. (2022). *Computer Vision Dan Pengolahan Citra Digital*. www.pustakaaksara.co.id
- Maryamah, M., Pratama, M. A., Erfit, M. R., Farhani, N. M., & Hartono, I. A. (2023). Klasifikasi Abjad SIBI (Sistem Bahasa Isyarat Indonesia) menggunakan Mediapipe dengan Metode Deep Learning. *Prosiding Seminar Nasional Sains Data*, 3(1), 134–141. <https://doi.org/10.33005/senada.v3i1.102>
- MediaPipe. (2023). <https://mediapipe.readthedocs.io/en/latest/solutions/hands.html>
- Nugroho, P. S., Purnami, N., Falerina, R., Perdana, R. F., Rahmadiyanto, Y., & So, C. V. (2021). Meningkatkan Kualitas Pendidikan Untuk Anak Dengan Berkebutuhan Khusus Tuna Rungu Dan Wicara Di Slb B Di Era Pandemi. *Jurnal Layanan Masyarakat (Journal of Public Services)*, 5(2), 364. <https://doi.org/10.20473/jlm.v5i2.2021.364-371>
- Oudah, M., Al-Naji, A., & Chahl, J. (2020). Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *Journal of Imaging*, 6(8). <https://doi.org/10.3390/JIMAGING6080073>
- Permana, A. A., S, W., Santoso, L. W., Wibowo, G. W. N., Wardhani, A. K., Rahmaddeni, Wahidin, A. J., Yuliasuti, G. E., Elisawati, Wijayanti, R. R., & Abdurrasyid. (2023). Machine Learning. In *Machine Learning* (Vol. 45, Issue 13). <https://books.google.ca/books?id=EoYBngEACAAJ&dq=mitchell+machine+learning+1997&hl=en&sa=X&ved=0ahUKEwiodmqfj8TkAhWGlkKHRCbAtoQ6AEIKjAA>
- Prasanda, A. (2022). *Mengenal Dua Jenis Bahasa Isyarat, BISINDO vs SIBI*. Translexi. <https://translexi.id/mengenal-dua-jenis-bahasa-isyarat-bisindo-vs-sibi/>
- Radya, M. (2024). *Penjelasan Mengenai Apa Itu Numpy Dan Fungsinya Pada Data Science*. Indobot Academy. <https://blog.indobot.co.id/penjelasan-mengenai-apa-itu-numpy-dan-fungsinya-pada-data-science/>
- Sari, A. R. (2021). *Sama-sama Bahasa Isyarat, Apa Perbedaan SIBI dan BISINDO?* Tempo.Co. <https://www.tempo.co/politik/sama-sama-bahasa-isyarat-apa-perbedaan-sibi-dan-bisindo--447992>
- Srimulia. (2022). *Mengenal OpenCV Dalam Python: Pengertian , Sejarah, Dukungan pada OS, Fitur-fitur*. Idmetaforia. <https://idmetafora.com/news/read/1177/Mengenal-OpenCV-Dalam-Python-Pengertian-Sejarah-Dukungan-pada-OS-Fitur-fitur.html>
- Wijoyo A, Saputra A, Ristanti S, Sya'ban S, Amalia M, & Febriansyah R. (2024). Pembelajaran Machine Learning. *OKTAL (Jurnal Ilmu Komputer Dan Science)*, 3(2), 375–380. <https://journal.mediapublikasi.id/index.php/oktal/article/view/2305>