Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.1, February 2026, pp. 729~736
e-ISSN: 2548-6861

Image Processing and Object Detection in the Indonesian Sign System (SIBI) for Hearing-Impaired Communication

Dewi Astria Faroek [1]*, Muhammad Yusuf [2]*, Haris [3]*
*Informatics Engineering, Universitas Muhammadiyah Sorong
dewiastriafaroek@um-sorong.ac.id [1], muhammadyusuf@um-sorong.ac.id [2], haris@gmail.com [3]

Article Info

Article history:

Received 2025-09-30
Revised 2025-12-24
Accepted 2026-01-07

Keyword:

Communcation,
Indonesian Sign Language (SIBI),
Computer Vision,
Real-Time Detection,
Web-Based Application

ABSTRACT

Communication is a fundamental human need, yet individuals with hearing impairments continue to face barriers due to limited access to sign language translation technologies. In Indonesia, the adoption of such technologies remains low, particularly in regions such as Sorong, Southwest Papua, creating a communication gap between the Deaf community and the general public. This study develops a web-based detection system for 36 classes of the Indonesian Sign System (SIBI) using the YOLOv5 algorithm. The dataset consists of 5,682 images of SIBI hand poses with variations in lighting and background, divided into 4,970 training images (87%), 376 validation images (7%), and 335 test images (6%). All data were processed through labeling, preprocessing, augmentation, balancing, and model training. The training was conducted for 150 epochs, and the evaluation results show that YOLOv5 is capable of detecting SIBI signs with significant accuracy. Performance evaluation using a confusion matrix achieved a detection accuracy of 95%, supported by stable precision and recall values and real-time inference performance on common web browsers. Usability testing with 20 respondents indicated satisfaction levels above 72.8%, demonstrating that the system is practical and easy to use. This research presents a validated real-time, web-based SIBI detection system that supports inclusive computer vision applications and enhances accessibility in public services such as education, healthcare, and administrative environments.

I. INRODUCTION

Communication is a basic need for every individual, but people with hearing impairments face limitations because they must rely on sign language as their primary means of interaction[1]. The World Health Organization (WHO, 2021) estimates that by 2050, 2.5 billion people will experience hearing loss, with 700 million of them requiring rehabilitation [2]. In Indonesia, according to the 2023 Indonesian Health Survey (SKI), the prevalence of hearing disability in individuals aged ≥ 1 year is 0.4% [3]. For children, data indicates that approximately 3 out of 100 children (3%) experience hearing loss (measured at ≥20 dB) according to a 2024 announcement by the Ministry of Health[4]. In Sorong City, Southwest Papua, the use of sign language translation technology is still very limited, characterized by a lack of data and low government attention[5][6]. This condition creates a communication gap between people with hearing impairments and the general public, necessitating technology-based innovations to bridge these limitations. The application developed in this study utilizes the YOLOv5s algorithm to detect 50 Indonesian Sign Language (BISINDO) movements thru a webcam camera. From the three training variations, the test results showed a maximum accuracy of 67% in the first test and 51.95% in the second test[7][8]. The results shown are less than optimal in supporting real-time detection. The model may also have difficulty generalizing if the training dataset is not sufficiently representative of real-world scenarios, thus affecting performance on new and real-time data.

The research conducted developed a SIBI gesture recognition system by combining hand skeleton and shape features using deep learning. The accuracy achieved is around 88%, but this study is still not optimal in handling variations in lighting conditions, dynamic gestures, and complex backgrounds[9]. Research related to image processing and machine learning also remains a challenge due to the varying speed of real-time movements[10][11]. However, the application of YOLO for the Indonesian Sign Language System (SIBI), especially web-based, is still very limited. Most current research focuses heavily on mobile or desktop applications, as well as static gesture prediction studies, highlighting the need for further research to develop a responsive, web-based real-time system accessible to the general public. This research aims to implement a web-based YOLOv5 algorithm for detecting and recognizing 36 classes of SIBI gestures.

The stages include image processing (labeling, resizing, grayscale, contrast adjustment, rotation, and dataset balancing) and system development using the Extreme Programming (XP) approach. The research results are expected to enrich academic studies in the field of computer vision, provide a practical system for inclusive communication for the hearing impaired, and serve as a technology-based policy reference in education and public services.

II. METHOD

This research uses handwritten image data consisting of the alphabet letters A–Z (26 classes) along with several additional words. The collected dataset was annotated using Roboflow as a tool for labeling and data augmentation, and then stored for further processing. The research process begins with collecting a dataset containing thousands of images of the Indonesian Sign System (SIBI). The dataset then undergoes a pre-processing stage to improve data quality through labeling, image resizing, augmentation, and balancing the number of images in each class.

After the initial processing is completed, the YOLO algorithm is developed to build a model capable of recognizing and detecting sign patterns with high accuracy. The flowchart illustrates the processing stages starting from dataset collection and labeling, followed by augmentation to increase the variation and quantity of training data. The processed dataset is then used to train the YOLO-based detection model, which is subsequently evaluated and stored as the final model to be implemented in a real-world system.

The development of both the model and the system application follows the Extreme Programming (XP) methodology, which includes iterative phases of planning, designing, coding, and testing to ensure that the resulting system is stable, responsive, and capable of being used in real time by users.

A. Image Processing

The preprocessing stage is carried out to improve the quality of the dataset prior to model training[12][13]. TThe techniques used include resizing to match YOLO input requirements, as well as data augmentation (rotation, flipping, zooming, and brightness adjustment) to increase data variability and reduce the risk of overfitting[14].

B. Development of the YOLO Algorithm

YOLOv5 is employed in this study due to its advantages in performing real-time object detection and delivering reliable accuracy performance [15][16]. The model development process includes:

- Modeling, i.e., adjusting the YOLO architecture according to classification requirements;
- Training and testing, with performance evaluation using accuracy, precision, recall, and mAP metrics;
- Storing the best-performing model for further implementation[17].

C. System Development Model

The Extreme Programming (XP) method is applied in system development, consisting of four main phases: planning for requirements identification, design for system architecture development, coding to implement YOLOv5 integrated with Roboflow, and testing to evaluate system performance using test data[18]. The Extreme Programming (XP) method is implemented through four primary phases: planning to identify system requirements, design to develop the system architecture and workflow, coding to implement the YOLOv5 algorithm integrated with Roboflow, and testing to evaluate the system's performance using testing data.

In the design phase, the web system architecture is built using a Flask-based client–server approach, where the user's browser sends HTTP requests to process images or camera streams. The server then processes these requests through a YOLOv5 pipeline consisting of four stages: (1) image preprocessing, (2) loading the YOLOv5 model using the best.pt weight file, (3) performing inference to detect gesture classes and bounding box coordinates, and (4) returning the detection results in JSON format to the client for direct visualization on the web interface[19]. This structured architecture ensures clear integration between the user interface, the Flask backend, and the YOLOv5 inference module, enabling the entire process to be reproduced consistently.

In the implementation phase, the system is designed to activate the camera, capture images in real-time, and process them using the YOLO-based detection procedure. When the system successfully detects a SIBI gesture, the detection results—consisting of the class label and bounding box—are displayed immediately, allowing users to obtain information quickly and accurately. The XP development model ensures that the system is built iteratively, remains responsive to user needs, and operates stably within real-world application environments. The Extreme Programming (XP) method is applied in system development, consisting of four main phases: planning for requirements identification, design for system architecture development, coding for implementing YOLOv5 integrated with Roboflow, and testing to evaluate system performance using test data[20][21]. The Extreme Programming (XP) method is applied in system development, consisting of four main phases: planning to identify system requirements, design to develop the system architecture and workflow, coding to implement the YOLOv5 algorithm integrated with Roboflow, and testing to evaluate system performance using test data[22]. During the implementation stage, the system is designed to activate the camera, capture images in real-time, and process them using a YOLO-based detection procedure. When the system successfully detects a SIBI sign, the detection results—consisting of the class label and bounding box—are displayed immediately, allowing users to obtain information quickly and accurately. The XP development model ensures that the system is built iteratively, remains responsive to user needs, and operates stably within real-world application environments.

III. RESULTS AND DISCUSSION

The dataset used in this study consists of 5,682 PNG/JPG images with 36 sign classes, which are divided into 4,970 images for training (87%), 335 images for testing (6%), and 376 images for validation (7%).

A. Development of the YOLO Algorithm

This study utilizes an SIBI sign image dataset in PNG/JPG format, collected through direct recording, independent data gathering, and downloads from Kaggle. The dataset was then processed in Roboflow to assign labels to each sign class.

B. Labeling with bounding boxes

At the labeling stage, the collected hand sign images are annotated by adding bounding boxes.

C. Split Dataset

At the dataset splitting stage, the annotated hand sign images are divided into train, test, and validation folders according to the model training requirements.

D. *Preprocessing* Data

Data preprocessing is performed to ensure that the dataset has consistent quality before being trained using the YOLOv5 algorithm. This process produces several outputs as follows:

1) *Resizing*: The image sizes, which were initially varied, were standardized to 416×416 pixels to match the YOLO model input requirements. This dimensional uniformity facilitates the training process and enhances detection stability.

2) *Augmentation:* The initial dataset consisting of N images was expanded with various augmentations:

- Rotation to simulate changes in hand angle;
- Horizontal flipping to capture both right- and left-hand gestures,
- Zooming in/out to enrich object distance variations,
- Brightness adjustment to adapt to different lighting conditions.

Through the augmentation process, the dataset size increased significantly to X images (approximately Y% growth from the original data). These additional variations play an important role in reducing the risk of overfitting and strengthening the model's generalization capability in real-world conditions.

E. Parameter Configuration

The configuration stage is carried out before the dataset training process begins, during which various essential hyperparameters—such as the number of epochs, batch size, and image resolution—are determined. The selection of these parameters greatly influences the model's performance because they define how many learning iterations are executed, how much data is processed at one time, and how images are fed into the model during training. With proper configuration, the training process can run more optimally, more stably, and in alignment with the system's requirements, as recommended in previous studies [23][24].

F. Training

The model training was conducted progressively using the available training data through iterative processes based on epochs and batches. In this context, one epoch represents a full cycle in which the model processes the entire dataset, allowing it to learn patterns comprehensively. Meanwhile, batches divide the dataset into smaller segments that are processed sequentially, enabling the training to run more efficiently and reducing computational load. This structured approach ensures that the model gradually improves its ability to recognize patterns with each iteration, ultimately enhancing accuracy and stability throughout the learning process.

G. Evaluation

The evaluation of the training results was conducted by analyzing various performance metrics, including the confusion matrix, precision curve, precision–recall curve, and recall curve[25]. These metrics collectively provide a comprehensive overview of the model's ability to accurately recognize and classify objects. The confusion matrix is used to observe the proportion of correct and incorrect predictions, allowing the identification of classes that are most frequently misclassified. The precision curve illustrates the model's accuracy in producing correct positive predictions across different confidence thresholds. The precision–recall curve offers a deeper perspective on the balance between precision and the model's capability to identify all positive objects, particularly in cases where the dataset is imbalanced. Meanwhile, the recall curve evaluates the system's consistency in correctly detecting objects at various threshold values. By considering all these metrics, the overall quality of the model can be assessed comprehensively, facilitating refinement and improvement of detection performance in subsequent development stages.

1) *Confusion Matrix*

The confusion matrix serves as an important evaluation tool to understand the model's performance in object detection, by showing the proportion of correct and incorrect detections. Correct detections are represented by True Positives and True Negatives, while errors are indicated by False Positives and False Negatives. To support training quality, the dataset was first subjected to preprocessing to ensure consistency with the YOLOv5 algorithm requirements. This process produced several outputs as follows:

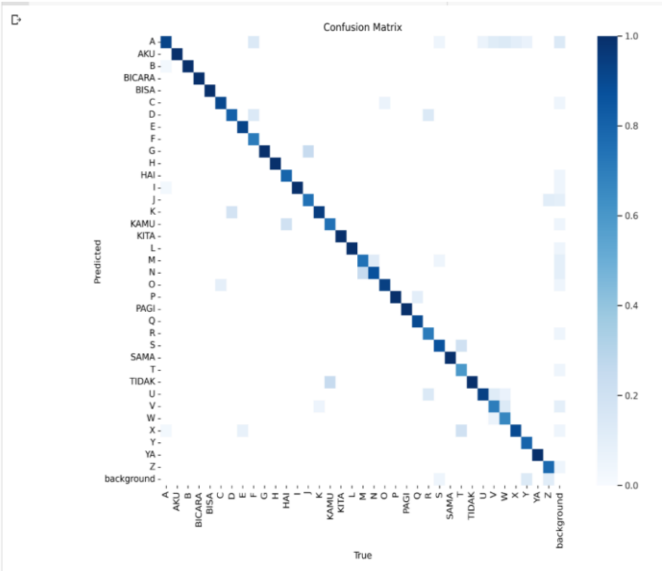

Figure 1. *Confusion Matrix*

Figure 1 presents the confusion matrix results obtained from model testing. Based on these results, accuracy, precision, and recall were calculated as the main indicators for assessing the model's effectiveness. The following are the computed accuracy values derived from the confusion matrix testing*:*

$$akurasi = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

$$akurasi = \frac{31{,}19 + 4{,}09}{31{,}19 + 1{,}06 + 0{,}61 + 4{,}09}$$

$$akurasi = \frac{35{,}28}{36{,}95}$$

$$akurasi = 0{,}95 \quad \text{atau } 95\%$$

Figure 2. *Accuracy*

2) *Precision Curve*

The precision curve illustrates how the precision value of a classification model changes with varying confidence thresholds used for data classification. Figure 3 presents the precision curve obtained after training.

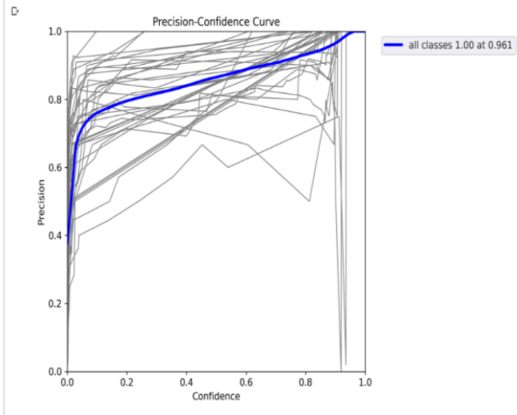

Figure 3. *Precission curve*

3) *Precission-Recall Curve*

In model evaluation, precision is used to assess the accuracy of positive detections, while recall measures the ability to identify all positive objects. As the threshold increases, precision improves but recall decreases, and this relationship is visualized in the precision–recall curve shown in Figure 4.

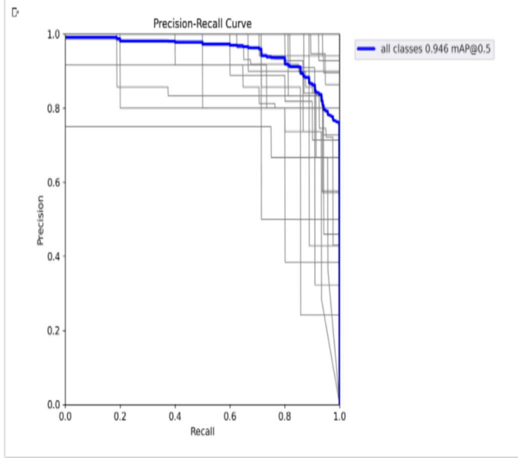

Figure 4. *Precission-Recall Curve*

4) *Recall Curve*

The recall curve graph helps in understanding the model's performance in detecting positive objects and in selecting the optimal confidence threshold according to application requirements. A good recall curve demonstrates high recall levels, steady and gradual improvements, slow decline, and provides clear insight into the trade-off between recall and precision. The recall curve results are shown in Figure 5.

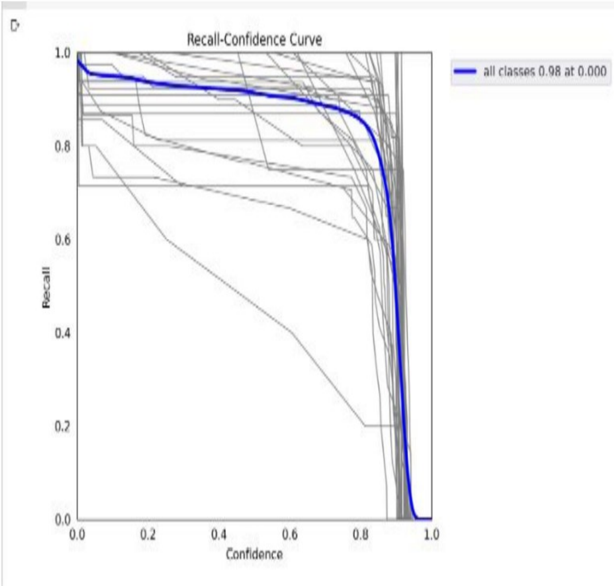

Figure 5. *Recall curve*

H. Detection Results

The results of the training process demonstrate a detection accuracy of over 90%, as shown in the following figure.

Figure 6. *Detection Results*

I. Save Model

The final stage of training produces a model that is subsequently saved in the form of a *best.pt* file, which serves as the primary reference used by the system to detect cues from input images. At this stage, the model parameters that yield the best performance during validation are automatically selected and stored to ensure optimal detection accuracy when the system is deployed. The *best.pt* file functions as the core component in the inference process, enabling the system to recognize and classify input signals consistently based on the knowledge it acquired throughout training. With this saved model, the system is prepared to operate in real-time detection scenarios while maintaining stability and high precision in identifying SIBI gestures.

J. System Development Model Extreme Programming

This stage establishes the objectives of developing the SIBI gesture detection system, including achieving a minimum detection accuracy of 90%. The Extreme Programming process is carried out through several integrated phases, beginning with the planning stage, in which a structured and goal-oriented strategy is formulated while still allowing room for evaluation and control to address discrepancies between the initial plan and actual development conditions. The design stage follows by translating the system architecture and interface logic into a technical blueprint that includes a lightweight web architecture built using Flask, where the client browser communicates with the server through HTTP requests, and the server processes uploaded images by forwarding them through a YOLOv5 inference pipeline. This pipeline consists of image preprocessing, model loading using the best.pt weight file, object detection inference, and returning a structured JSON output to the client for real-time visualization.

During the coding stage, the system is developed iteratively and adaptively based on the Extreme Programming approach, enabling rapid responses to changing requirements while ensuring efficient backend processing through the lightweight yet reliable Flask environment. With this combination of a fast development methodology and appropriate technology choices, the system is able to operate optimally in accordance with its intended objectives.

In the testing phase, blackbox testing shows that all core functionalities—from image upload and model processing to the display of detection results—operate according to specifications without any functional errors[26][27]. Tests were conducted to identify potential errors, and the results are presented in Table 1. Meanwhile, usability testing indicates that users can operate the system easily, the interface is quickly understood, response times are adequate, and user satisfaction levels are categorized as high. These combined testing results ensure that the system not only functions correctly but also provides an effective, user-friendly, and reliable experience in real-world scenarios.

TABLE I
BLACKBOX TESTING

No	Testing Activity	Test Results	Conclusion
1	Webcam Object Detection	Active Webcam	Valid
2	Gesture Detection	Displays Gesture Detection by class	Valid

Source: processed data, 2025

Usability testing also indicated that the system is easy to operate, the interface is quickly understood, the response time is considered adequate, and the overall user satisfaction level falls into the high category[28]. Questionnaire data obtained from 20 respondents were analyzed using a Likert scale, consisting of 15 respondents from the deaf community in Sorong City and 5 respondents from the general public. This analysis further reinforces the finding that the system not only functions correctly but also provides an effective, easy, comfortable, and reliable user experience in real-world scenarios. then the calculation results are presented in the following table 2:

TABLE 2
USEBILITY TESTING

Answer scale	Description	Score
SA	Strongly Agree	5
A	Agree	4
FA	Fairly Agree	3
D	Disagree	2
SD	Strongly Disagree	1

The results of calculating the percentage of the total respondent score can be seen as follows:

$$Y = \frac{x}{Ideal\ score} x\ 100$$

Y = The desired percentage value (%)

X = The sum of the answer category values multiplied by the frequency (∑ 〖=N x R〗)

N = The value of each answer

R = Frequency

Ideal score = The highest value multiplied by the number of samples (5x20 = 100)After obtaining the Y value, the next step is to find the interval (range) and percentage interpretation using the formula below:

I = 100/Total Score (Likert)

Therefore, I = 100/5 = 20

Result (I) = 20

(this is the interval from the lowest 0% to the highest 100%). Score each question in the following table:

TABLE 3
QUESTION SCORE

Question 1	Description	Score (N)	Respondent	NxR
The application is running well	Strongly Agree	5	0	0
	Agree	4	14	56
	Fairly Agree	3	6	18
	Disagree	2	0	0
	Strongly Disagree	1	0	0
Total			**20**	**74**
Percent $Y = \frac{74}{100}\ x\ 100$				74%
Question 2	**Description**	**Score (N)**	**Respondent**	**NxR**
The application has an easy to understand and easy to use display.	Strongly Agree	5	0	0
	Agree	4	10	40
	Fairly Agree	3	10	30
	Disagree	2	0	0
	Strongly Disagree	1	0	0
Total			**20**	**70**
Percent $Y = \frac{70}{100}\ x\ 100$				70%
Question 3	**Description**	**Score (N)**	**Respondent**	**NxR**
Useful Applications for Society	Strongly Agree	5	2	10
	Agree	4	12	48
	Fairly Agree	3	4	12
	Disagree	2	2	4
	Strongly Disagree	1	0	0
Total			**20**	**74**
Percent $Y = \frac{74}{100}\ x\ 100$				**74%**
Question 4	**Description**	**Score (N)**	**Respondent**	**NxR**
The SIBI sign language application can help users understand the SIBI sign language detection system.	Strongly Agree	5	4	20
	Agree	4	10	40
	Fairly Agree	3	2	6
	Disagree	2	4	8
	Strongly Disagree	1	0	0
Total			**20**	**74**
Percent $Y = \frac{74}{100}\ x\ 100$				**74%**
Question 5	**Description**	**Score (N)**	**Respondent**	**NxR**
Application usage is satisfactory	Strongly Agree	5	2	10
	Agree	4	12	48
	Fairly Agree	3	2	6
	Disagree	2	4	8
	Strongly Disagree	1	0	0
Total			**20**	**72**
Percent $Y = \frac{72}{100}\ x\ 100$				72%
Mean total				**72,8%**

Source: processed data, 2025

The usability testing results of the SIBI sign language detection application indicate that the system functions well, is easy to use, beneficial for the community, and provides a relatively high level of user satisfaction. Based on five evaluation indicators assessed by 20 respondents using a Likert scale, an average score of 72.8% was obtained, which falls into the ‘good’ category, with responses predominantly in the ‘Agree’ and ‘Strongly Agree’ categories. The highest scores were recorded in the aspects of application functionality and societal benefits (74%), while interface simplicity and ease of use received 70%. Overall, the application was considered effective in detecting SIBI gestures, assisting users in understanding the system, and delivering a satisfactory user experience. These findings demonstrate that the YOLOv5-based application is feasible as an inclusive communication tool and holds strong potential for further development, particularly in enhancing the user interface and optimizing performance to be more adaptive to diverse real-world conditions.

IV. Conclusions

This study demonstrates the successful implementation of the YOLOv5 algorithm for web-based detection of SIBI sign language gestures by utilizing a dataset of 5,682 images from 36 sign classes that underwent labeling, dataset splitting, and preprocessing through resizing, augmentation, and balancing to improve data quality. The trained model achieved a detection accuracy of 95% with consistent precision and recall values, as well as stable real-time inference performance. System development using the Extreme Programming (XP) method enabled a structured process through the planning, design, coding, and testing stages. Black-box testing confirmed that all system functions operated according to specifications, while usability testing involving 20 respondents produced an average score of 72.8% (good category), indicating that the application is easy to use, beneficial, and provides a satisfying user experience. Thus, this study affirms that the YOLOv5-based SIBI gesture detection system is feasible as an inclusive communication tool, offering advantages in speed and real-time accuracy.

The sustainability of this research presents broad development opportunities, one of which is expanding the model to support Bahasa Isyarat Indonesia (BISINDO), which is more commonly used across various regions, thereby enhancing accessibility and inclusivity for the deaf community. Furthermore, the system can be optimized through multimodal integration, such as combining camera input with text or voice output, which would enrich interactions and create a more natural communication experience. This multimodal integration has the potential to produce a more intelligent and adaptive sign detection system capable of providing more comprehensive communication support in diverse real-world scenarios. With continuous development, this system is expected to evolve into an increasingly effective and relevant assistive technology that supports the communication needs of the deaf community in Indonesia.

Acknowledgement

We would like to express our gratitude to Universitas Muhammadiyah Sorong, our place of service, and to the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia for funding support under the 2025 Regular Beginner Lecturer Research (PDP) scheme. Our deepest appreciation is also extended to all parties who contributed to the smooth execution and completion of this research.

References

[1] S. M. Fitriyani, Lathifah Qurrotu Ainii, Raudhatul Jannah, “Analysis of Sign Language Skills in Improving Communication and Learning for Deaf Children,” vol. 5, no. 1, pp. 167–186, 2021.

[2] B. O. Olusanya, A. C. Davis, and H. J. Hoffman, “Hearing loss grades and the International classification of functioning, disability and health.,” *Bull. World Health Organ.*, vol. 97, no. 10, pp. 725–728, Oct. 2019, doi: 10.2471/BLT.19.230367.

[3] Survey kesehatan indonesia (Ski), “Survei Kesehatan Indonesia 2023 (SKI),” *Kemenkes*, p. 235, 2023.

[4] R. Zaskia, R. Khairunnisa, and L. S. Fernanda, “Hubungan Gangguan Pendengaran Dengan Kemampuan Berbahasa Pada Anak – Anak,” *J. Skripta*, vol. 11, no. 1, pp. 54–59, 2025, doi: 10.31316/skripta.v11i1.7185.

[5] Heriyanti, Jeane Susan Tapada, Kris Uluelang, Putri Jalila Umasugi, and Isnaeni Wahab, “Teacher’S Strategies in Teaching English To Deaf Students in a Special School Sorong,” *JLE J. Lit. English Educ. Study Progr.*, vol. 4, no. 02, pp. 91–101, 2023, doi: 10.47435/jle.v4i02.2332.

[6] Adam Nurmansyah, Nanda Rizqia Rhamadhani, Sabrina Alfarissy Nur Hakim, Sri Azhari Agustin, and Siti Hamidah, “Permasalahan Komunikasi Yang Kerap Terjadi Pada Penyandang Disabilitas,” *J. Pendidikan, Bhs. dan Budaya*, vol. 2, no. 2, pp. 200–210, 2023, doi: 10.55606/jpbb.v2i2.1515.

[7] 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.

[8] T. Pramanik, P. G. Burade, and S. Sharma, “Analysis of real-time multi-surveillance detection model using YOLO v5,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 38, no. 3, p. 1634, 2025, doi: 10.11591/ijeecs.v38.i3.pp1634-1641.

[9] A. M. Hidayahtullah, “Sistem Deteksi Simbol Pada SIBI (Sistem Isyarat Bahasa Indonesia) Secara Realtime menggunakan Mobilenet-SSD,” *dinamika*, vol. 33, no. 1, pp. 1–12, 2022.

[10] M. Ugale, O. R. A. Shinde, K. Desle, and S. Yadav, *A Review on Sign Language Recognition Using CNN*. Atlantis Press International BV, 2023. doi: 10.2991/978-94-6463-136-4_23.

[11] L. Sandy, A. Putra, R. R. Yacoub, and E. Kusumawardhani, “SIBI Introduction Using YOLOv8 as Android-Based Learning Media,” vol. 3, no. 1, pp. 24–33, 2025, doi: 10.26418/telectrical.v3i1.93519.

[12] D. Luthfy, C. Setianingshi, and M. W. Paryasto, “Indonesian Sign Language Classification Using You Only Look Once,” *eProceedings Eng.*, vol. 10, no. 1, pp. 454–459, 2023.

[13] W. Saputro and D. B. Sumantri, “Klasifikasi Citra Dalam Gerak Tangan Bahasa Isyarat Sibi Menggunakan Algoritma K-NN,” *INTECOMS J. Inf. Technol. Comput. Sci.*, vol. 5, no. 2, pp. 180–188, 2022, doi: 10.31539/intecoms.v5i2.4446.

[14] B. K. Pratama, Sri Lestanti, and Yusniarsi Primasari, “Implementasi Algoritma You Only Look Once (YOLO) untuk Mendeteksi Bahasa Isyarat SIBI,” *ProTekInfo(Pengembangan Ris. dan Obs. Tek. Inform.*, vol. 11, no. 2, pp. 7–14, 2024, doi: 10.30656/protekinfo.v11i2.9105.

[15] R. Hesananda, I. A. Noviani, and M. Zulfariansyah, “Implementasi YOLOv5 untuk Deteksi Objek Mesin EDC: Evaluasi dan Analisis,” *BIOS J. Teknol. Inf. dan Rekayasa Komput.*, vol. 5, no.

2, pp. 104–110, 2024, doi: 10.37148/bios.v5i2.127.
[16] Salsabilla Azahra Putri, Murinto, and Sunardi, “Implementasi Algoritma YOLOv5 untuk Otomatisasi Iklan Layananan Publik tentang Larangan Merokok,” *J. Inform. Polinema*, vol. 11, no. 2, pp. 195–202, 2025, [Online]. Available: https://jurnal.polinema.ac.id/index.php/jip/article/view/6343
[17] M. Vicko, P. Ardiansyah, J. Sahertian, and R. Heri, “Penerapan Algoritma YOLO dalam Sistem Klasifikasi Kendaraan,” vol. 9, pp. 1151–1157, 2025.
[18] P. Explained and K. Beck, *Praise for Extreme Programming Explained , Second Edition*.
[19] J. Redmon, “YOLOv3: An Incremental Improvement”.
[20] R. Rudianto, “Penerapan Metode Extreme Programming Dalam Pembangunan Aplikasi Sistem Penunjang Keputusan,” *J. Sist. Inf. dan Bisnis Cerdas*, vol. 16, no. 1, pp. 21–30, 2023, doi: 10.33005/sibc.v16i1.6.
[21] R. Dwiyanto, D. W. Widodo, and P. Kasih, “Implementasi Metode You Only Look Once (YOLOv5) Untuk Klasifikasi Kendaraan Pada CCTV Kabupaten Tulungagung,” *Semin. Nas. Inov. Teknol.*, vol. 1, no. 1, pp. 102–104, 2022.
[22] A. Pambudi and W. Apriandari, “An Extreme Programming Approach for Instructor Performance Evaluation System Development,” vol. 8106, pp. 126–135, 2023.
[23] A. E. Kalambia, G. Kevin, and P. H. Saputro, “Model Deep Learning YOLOv5 untuk Identifikasi Cuaca: Cloudy, Rain, Shine, dan Sunrise,” *Inform. J. Ilmu Komput.*, vol. 20, no. 3, pp. 125–132, 2024, doi: 10.52958/iftk.v20i3.11089.
[24] M. H. Ashar and D. Suarna, “KLIK: Kajian Ilmiah Informatika dan Komputer Implementasi Algoritma YOLOv5 dalam Mendeteksi Penggunaan Masker Pada Kantor Biro Umum Gubernur Sulawesi Barat,” *Media Online*, vol. 3, no. 3, pp. 298–302, 2022, [Online]. Available: https://djournals.com/klik
[25] R. Hesananda, “Implementasi Model Yolo V5 Untuk Deteksi Korek Api Dalam Keamanan Penerbangan,” *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 1, 2025, doi: 10.23960/jitet.v13i1.5553.
[26] A. Jailani and M. A. Yaqin, “Pengujian Aplikasi Sistem Informasi Akademik menggunakan Metode Blackbox dengan Teknik Boundary Value Analysis,” *J. Autom. Comput. Inf. Syst.*, vol. 4, no. 2, pp. 60–66, 2024, doi: 10.47134/jacis.v4i2.78.
[27] Y. D. Wijaya and M. W. Astuti, “Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions,” *J. Digit. Teknol. Inf.*, vol. 4, no. 1, p. 22, 2021, doi: 10.32502/digital.v4i1.3163.
[28] A. A. Santika, T. H. Saragih, and M. Muliadi, “Penerapan Skala Likert pada Klasifikasi Tingkat Kepuasan Pelanggan Agen Brilink Menggunakan Random Forest,” *J. Sist. dan Teknol. Inf.*, vol. 11, no. 3, p. 405, 2023, doi: 10.26418/justin.v11i3.62086.