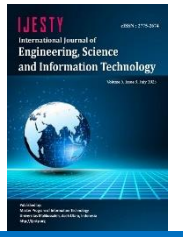
International Journal of Engineering, Science and InformationTechnology
Volume 5, No. 3 (2025) pp. 285-294
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i3.948
Research Paper, Short Communication, Review, Technical Paper



Optimizing YOLO-Based Algorithms for Real-Time BISINDO Alphabet Detection Under Varied Lighting and Background Conditions in Computer Vision Systems

Lilis Nur Hayati[1,3], Anik Nur Handayani[1*], Wahyu Sakti Gunawan Irianto[1], Rosa Andrie Asmara[2], Dolly Indra[3] , Nor Salwa Damanhuri[4]

[1]Department of Electrical Engineering and Informatics, Universitas Negeri Malang, Malang, Indonesia
[2]Department of Information Technology, State Polytechnic of Malang, Malang, Indonesia
[3]Department of Computer Science, Universitas Muslim Indonesia, Makassar, Indonesia
[4]Electrical Engineering Studies, Universiti Teknologi MARA, Malaysia

**Corresponding author Email: aniknur.ft@um.ac.id*

The manuscript was received on 16 January 2025, revised on 28 February 2025, and accepted on 1 June 2025, date of publication 17 June 2025

Abstract

This research explores the optimization of YOLO-based computer vision algorithms for real-time recognition of Indonesian Sign Language (BISINDO) letters under diverse environmental conditions. Motivated by the communication barriers faced by the deaf and hearing communities due to limited sign language literacy, the study aims to enhance inclusivity through advanced visual detection technologies. By implementing the YOLOv5s model, the system is trained to detect and classify correct and incorrect BISINDO hand signs across 52 classes (26 correct and 26 incorrect letters), utilizing a dataset of 3,900 images augmented to 10,920 samples. Performance evaluation employs k-fold cross-validation (k=10) and confusion matrix analysis across varied lighting and background scenarios, both indoor and outdoor. The model achieves a high average precision of 0.9901 and recall of 0.9999, with robust results in indoor settings and slight degradation observed under certain outdoor conditions. These findings demonstrate the potential of YOLOv5 in facilitating real-time, accurate sign language recognition, contributing toward more accessible human-computer interaction systems for the deaf community.

Keywords: *YOLOv5, BISINDO, Sign Language Recognition, Object Detection, Real-Time Detection.*

1. Introduction

Communication is one of the most essential needs in human life. Through communication, people can understand the intentions and goals of others, from sharing ideas to sending and receiving information [1]. However, it is different for those with physical limitations. People with hearing impairments usually communicate using sign language.

Sign language is a form of non-verbal human communication expressed through hand movements to convey information to the recipient visually. It is commonly used by people who are deaf. The term "deaf" is generally used to describe hearing difficulties ranging from mild to severe, classified into deafness and hard of hearing [2]. One of the most effective types of sign language—not only limited to the deaf community but also accessible to everyone—is Indonesian Sign Language (BISINDO) [3].

BISINDO is a method of sign language used by individuals with hearing impairments, utilizing hand gestures, facial expressions, and body movements to form symbols representing letters or words. BISINDO is supported by the Indonesian Deaf Welfare Movement (GERKATIN) and was developed by the deaf community itself [4].

One of the major issues today is that most hearing people do not learn or understand how to use sign language. As a result, communication between hearing individuals and the deaf remains difficult. Therefore, it is necessary to develop a system that can easily recognize and translate an image or video into alphabetic text [5]. The application of the You Only Look Once (YOLO) algorithm for detecting BISINDO letters using Computer Vision focuses on recognizing letter similarities and improving precision in real time [6],[7]. YOLO is an object detection algorithm that divides an image into a grid system, where each grid cell is responsible for detecting objects within it [8],[9]. YOLO is one of the most well-known object detection algorithms due to its speed and accuracy [10].

Based on previous research by Steve Daniels et al., titled *"Indonesian Sign Language Recognition using YOLO Method,"* an Indonesian Sign Language recognition system using YOLO was implemented. Their image and video data experiments achieved 100% and 72.97% accuracy, respectively [11]. Another study by Miguel Rivera-Acosta et al., titled *"Spelling Correction Real-Time American Sign*

Language Alphabet Translation System Based on YOLO Network and LSTM," successfully translated 24 ASL alphabet signs and two additional signs in real-time with a mAP@50 of 99.81% and a frame processing rate of 61.35 [12]. In comparison, research conducted by Mohamad Amar Mustaqim Mohamad Asri et al., titled *"A Real-Time Malaysian Sign Language Detection Algorithm Based on YOLOv3,"* showed less promising results, with a detection consistency and identification accuracy of 63.06% during training and 72% in system implementation [13].

2. Literature Review

Various studies have utilized the YOLO algorithm and deep learning techniques for visual recognition in different contexts, including sign language recognition [14],[15]. The license plate detection system using YOLO and Tesseract OCR successfully identified vehicle plates in both portrait and landscape orientations under sufficient lighting at a confidence threshold of 0.5 [16]. Life sign detection in disaster victims using YOLO and OpenPose achieved optimal results with a training configuration of 90%:10% data partition, 0.001 learning rate, batch size of 64, and 1000 max batches [17]. Hand gesture detection based on image processing reached high accuracy levels—up to 98% across different background colours. A combination of YOLO and AlexNet achieved 100% classification accuracy for hand gestures under varying scales, orientations, lighting conditions, and complex backgrounds [18]. In comparison, the YOLOv3 model demonstrated strong performance with 97.68% accuracy and an F1-score of 96.70% [10]. In the context of BISINDO, a CNN-based system achieved 76% prediction accuracy, outperforming LeNet (19%) and AlexNet (60%) in real-time hand gesture recognition [19].

3. Research Methods

3.1. BISINDO

Sign language is a method of communication that uses symbols without the use of voice, often referred to as *non-verbal communication*. The symbols used involve hand movements and other body parts, such as facial expressions, images, symbols, or gestures that carry specific meanings and can be understood by both the speaker and the receiver [12]. **BISINDO** (Indonesian Sign Language) is one of two sign languages used in Indonesia, alongside SIBI (Indonesian Sign System). The BISINDO alphabet consists of 26 characters, from A to Z. Some letters can be formed using one hand, such as C, E, I, J, L, O, R, U, V, and Z, while others require two hands to form, including A, B, D, F, G, H, K, M, N, P, Q, S, T, W, X, and Y.

Fig 1. BISINDO Alphabet [20]

3.2. You Only Look Once (YOLO)

YOLO is an open-source object detection algorithm that prioritizes object detection speed and efficiency [21]. The YOLO model was introduced in 2016. Its working principle involves dividing an image into an S x S grid of cells before predicting objects. If the centre point of an object falls within a grid cell, that grid cell is responsible for detecting the corresponding object. Each grid cell predicts a bounding box and a confidence score [22]. The latest YOLO model is **YOLOv5**, developed by Joseph Redmon, one of the founders of Ultralytics LLC, in 2020. YOLOv5 offers several advantages over previous generations of YOLO. These include a more user-friendly PyTorch framework and easier training of custom datasets compared to the last DarkNet-based framework [14].

3.3. Confusion Matrix

The confusion matrix is a performance evaluation tool for prediction methods that calculates the accuracy of the classification process [16]. The ratio of correct predictions to the total number of data determines accuracy. Precision is the ratio of accurate optimistic predictions to the total number of positive predictions. Recall is obtained from the ratio of accurate optimistic predictions to the total number of positive cases. In a confusion matrix, there are four possible outcomes when detecting objects: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). The formulas for calculating Accuracy, Precision, and Recall using the confusion matrix are as follows (1)–(3):

$$\text{Accuracy} = (TP + TN) / (TP + FP + FN + TN) \quad (1)$$

$$\text{Precision} = TP / (TP + FP) \quad (2)$$

$$\text{Recall} = TP / (TP + FN) \quad (3)$$

3.4. K-Fold Cross Validation

K-Fold Cross Validation is a statistical method used to evaluate the performance of a designed model or algorithm [23]. The dataset is divided into training and validation data in the training phase. The model is trained using the training data and validated using the validation data for K number of folds [24].

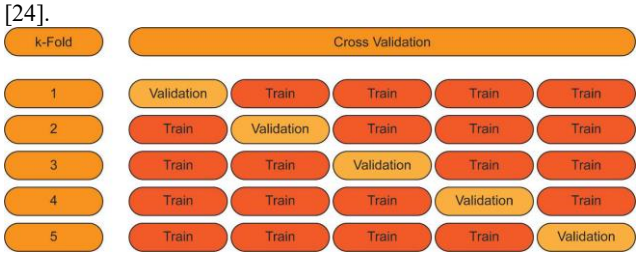


Fig 2. K-Fold Cross Validation [24]

In K-Fold Cross Validation, the training data is divided into K-1 for iteration 1, K-2 for iteration 2, and so on. The calculation formula depends on the type of cross-validation used. In this study, **precision and recall** formulas are applied to calculate the accuracy of the detection model.

In this study, we propose a BISINDO alphabet recognition system, as depicted in Fig 3. This section comprises both the training and testing phases [25]. The training stage involves input, pre-processing, augmentation, generation, and the dataset, which will be explained in the following sections.

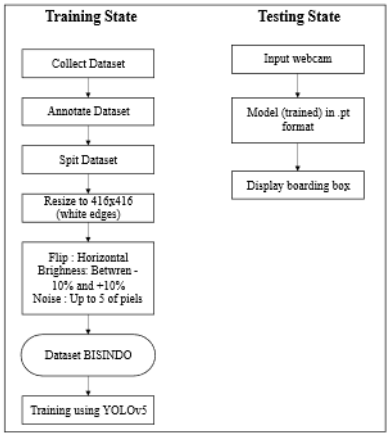


Fig 3. Flowchart of the Research Design

3.5. Input Image Data

The image data used in this research consists of the BISINDO alphabet. In the BISINDO alphabet, there are 26 letters, where some letters can be formed using a single hand, such as "C, E, I, J, L, O, P, R, U, V, and Z," and others using both hands, including "A, B, D, G, H, K, M, N, P, Q, S, T, W, X, and Y," [26]. The first process is to collect data from 3900 datasets. The dataset is the BISINDO A-Z True and False alphabets taken using a smartphone camera with an aspect ratio of 4:3 and a background of black, white and red clothes with a white wall.

3.6. Dataset Annotation

The next step is annotation or labelling, which is performed using RoboFlow. In this stage, the author labelled a total of 3,900 images consisting of 26 "Correct BISINDO A–Z" classes and 26 "Incorrect BISINDO A–Z" classes, each containing 75 images. Labelling involves providing specific information about the location and type of objects in the pictures. This annotation information is essential for training the model to recognize and detect objects within the images [27].

3.7. Dataset Splitting

After annotation, the dataset is split into training and validation sets. The training set consists of 3,510 images, which were augmented to become 10,350, while the validation set contains 390 photos. Dataset splitting is conducted to train and evaluate the model, prevent overfitting, and assess the model's performance.

3.8. Pre-Processing

In this stage, a pre-processing step is carried out by resizing the images to a smaller dimension to improve model training speed—from the original size of 3264x1840 to 416x416 (fit with white edges). This pre-processing was performed using RoboFlow [13].

3.9. Augmentation

After pre-processing, dataset augmentation is applied to the training data, increasing the number of images from 3,900 to 10,920. The augmentations include horizontal flipping, brightness adjustment (between -10% and +10%), and noise addition (up to 5% of pixels). The augmentation step aims to increase the number of samples, enhance variability, improve model reliability, and reduce overfitting. A dataset generation process is then conducted on RoboFlow to store the results of dataset annotation, splitting, pre-processing, and augmentation into the RoboFlow database.

3.10. Training Using Yolov5

In this stage, the author performs model training using the YOLOv5 PyTorch format on the BISINDO dataset to evaluate model performance in Google Colab [28]. Several arguments are set during training, including image size, batch size, number of epochs, and weights. The training process results include evaluation outputs in .png format, such as the confusion matrix, precision-confidence, recall-confidence, F1-confidence, and precision-recall graphs. The final trained model is saved as a .pt file and uploaded to Google Drive for real-time testing using Anaconda Prompt.

3.11. Webcam Input

Webcam input is used during the system testing phase, utilizing the DroidCam application connected to a smartphone and a laptop camera to perform real-time testing.

3.12. Trained Model

At this stage, YOLOv5 loads the trained model in .pt format and processes the input image data from the webcam to perform object prediction and detection [29].

3.13. Displaying Bounding Boxes

After detection, the system displays bounding boxes that include each detected object's class and confidence score. YOLOv5 continuously performs prediction and detection based on the webcam input, allowing real-time object detection.

The YOLOv5 algorithm with the YOLOv5s variant works as follows:

1. The input image is divided into a 13x13 grid.
2. Each grid cell predicts three bounding boxes, each with attributes including x and y coordinates, width, height, and a confidence score.
3. The model predicts object classes for each bounding box using the softmax technique.
4. After processing all grid cells, the model applies non-maximum suppression (NMS) to eliminate overlapping bounding boxes that refer to the same object. This technique selects the bounding box with the highest confidence score.
5. The final output of the model is a list of selected bounding boxes along with the corresponding object classes and confidence scores.
6. The last step is to draw the bounding boxes on the input image and display the object detection results on the screen.

4. Result and Discussion

4.1. Model Evaluation

The model evaluation was carried out using the 10-fold cross-validation method, where the dataset used was the result of splitting the data into training and validation sets [27]. The evaluation process involved 10 training iterations to obtain the Average Precision (AP) and Average Recall (AR) values. This method provides diverse insights into the model's performance; cross-validation was applied to give a more comprehensive evaluation. The evaluation results for each iteration are presented in Table 1.

Table 1. Cross-validation K=10

Cross-validation	precision	recall
Iterasi 1	0.99012	1
Iterasi 2	0.9904	0.99969
Iterasi 3	0.99016	1
Iterasi 4	0.99005	1
Iterasi 5	0.98988	1
Iterasi 6	0.99001	1
Iterasi 7	0.99013	1
Iterasi 8	0.99018	0.99961
Iterasi 9	0.99089	0.99978
Iterasi 10	0.98949	1

Based on Table 1, by performing cross-validation, the average values obtained were an **Average Precision** of **0.990131** or **99%** and an **Average Recall** of **0.999908** or **99%**. The evaluation results indicate the model has excellent performance; however, there is room for improvement. Researchers may enhance the model further to improve its performance in the future.

4.2. Model Testing

The testing of this system was conducted in real-time using an Android camera and a laptop, involving both self-testing and testing with other actors [30]. Each actor wore different coloured clothing and accessories. The tests were carried out in two conditions: **indoor** and **outdoor** environments. The evaluation results from the testing phase were analyzed using a **confusion matrix**, which provided values for **accuracy**, **precision**, and **recall**. The following are examples of terms used to describe the prediction results of a model or algorithm.

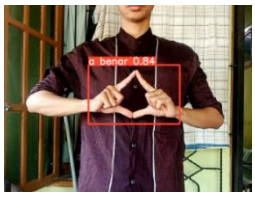


Fig 4. True Positive (TP)

Fig 5. False Positive (FP)

Fig 6. True Negative (TN)

Fig 7. False Negative (FN)

Fig 4 True Positive (TP) is when the hand correctly forms the letter "a," and the system accurately predicts it as the correct letter. As shown in Fig 4, the class predicted by the model is "correct." Fig 5 False Positive (FP) is when the hand forms the letter "t correctly," but the system fails to predict it accurately. As illustrated in Fig 5, the class predicted by the model is "x incorrect." Fig 6 True Negative (TN) is when no letter is formed, and the system does not display a bounding box correctly. Fig 7 False Negative (FN) is when a letter is formed, but the system fails to display the corresponding bounding box. As shown in Fig 7, when the hand correctly forms the letter "m," the system cannot display a bounding box labelled "m correct."

4.3. Indoor Testing

The testing in this category was conducted indoors, with a testing distance of approximately ±70 cm and the camera positioned at chest height. The light intensity was around ±10 LUX, measured using the Lux Light Meter application. The author wore red clothing, a striped shirt, and a long-sleeved maroon top, with a white cloth background used for certain letters such as "Z," and accessories including a wristwatch and rings. Meanwhile, the other actor wore black clothing with a white wall as the background for specific letters like "Z."

The test results were manually recorded one by one from the letter A to Z. The following are examples of indoor testing using accessories with both hands and one hand.

Fig 8. Testing with accessories (an correct)

Fig 9. Testing with accessories (an Incorrect)

The **confusion matrix for Test 1 using accessories in an indoor setting** is presented in **Table 2**.

Table 2. Confusion Matrix Testing (Correct)

Correct Class	Indoor testing using accessories				Testing with a Striped Shirt in an Indoor Setting				Testing with Another Actor in an Indoor Setting				Long-sleeve testing indoors				Indoor Testing with Laptop Camera			
	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
A Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
B Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
C Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
D Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
E Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
F Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
G Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
H Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
I Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
J Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
K Correct	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
L Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
M Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
N Correct	1	1	1	0	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0
O Correct	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	1	0
P Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Q Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
R Correct	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0
S Correct	1	1	0	0	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
T Correct	1	1	0	0	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
U Correct	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
V Correct	1	1	0	0	0	1	0	1	1	1	1	0	1	1	0	0	1	1	0	0
W Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
X Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Y Correct	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
Z Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Total	**26**	**26**	**1**	**0**	**23**	**26**	**6**	**1**	**26**	**26**	**2**	**0**	**26**	**26**	**0**	**0**	**26**	**26**	**0**	**0**

Table 3. Confusion Matrix IndorTesting (Correct)

Incorrect Class	Indoor testing using accessories				Testing with a Striped Shirt in an Indoor Setting				Testing with Another Actor in an Indoor Setting				Long-sleeve testing indoors				Indoor Testing with Laptop Camera			
	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
A Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
B Incorrect	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0
C Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0
D Incorrect	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0
E Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
F Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
G Incorrect	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
H Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
I Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
J Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0
K Incorrect	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
L Incorrect	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
M Incorrect	1	1	0	0	0	1	0	1	1	1	0	0	1	1	0	0	1	1	0	0
N Incorrect	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
O Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
P Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0

Q Incorrect	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0
R Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
S Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
T Incorrect	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
U Incorrect	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
V Incorrect	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0
W Incorrect	1	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0
X Incorrect	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0
Y Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Z Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Total	**26**	**26**	**4**	**0**	**25**	**1**	**5**	**1**	**26**	**26**	**4**	**0**	**26**	**26**	**2**	**0**	**26**	**26**	**2**	**0**

Based on Table 2 and Table 3 above, all classes have **True Positive** and **True Negative** values. In contrast, the classes **B Incorrect**, **L Incorrect**, **N Incorrect**, and **W Incorrect** have **False Positive** values. The following is an example image of testing using a striped shirt with both hands in an indoor setting.

Fig 10. Testing with striped shirt (a correct)

Fig 11. Testing with striped shirt (incorrect)

The confusion matrix for **Test 2 using a striped shirt in an indoor setting** is presented in **Table 3**.
Based on Table 2 and Table 3 above, 48 classes have **True Positive** values, and 52 have **True Negative** values. In contrast, the classes D Incorrect, G Incorrect, K Correct, K Incorrect, O Correct, S Correct, T Correct, T Incorrect, U Correct, U Incorrect, and Y Correct have False Positive values. **M Incorrect** has a **False Negative** value. The following is an image of testing with another actor using both hands indoors.

Fig 12. Testing with another actor (a correct)

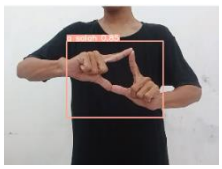


Fig 13. Testing with another actor (incorrect)

Based on Table 2 and Table 3 above, all classes have **True Positive** and **True Negative** values. In contrast, the classes **Q Incorrect**, **R Correct**, **V Correct**, **V Incorrect**, **W Incorrect**, and **X Incorrect** have **False Positive** values. This is an example of a test image involving a subject wearing long sleeves and using both hands indoors.

Fig 14. Long-sleeve testing (a correct)

Fig 15. Long-sleeve testing (an incorrect)

The following is the confusion matrix for Test 4 using long-sleeved clothing indoors, as shown in Table 5.
According to Table 2 and Table 3, all classes were categorized as True Positive and True Negative, except for C and D (Incorrect) and class N (Correct), which were classified as False Negative. This is an example of a test image captured using a laptop camera featuring two-handed gestures performed indoors.

Fig 16. Laptop camera testing (a correct)

Fig 17. Laptop camera testing (an incorrect)

The following is the confusion matrix for Test 5 using a laptop camera indoors, as shown in Table 6.
According to Table 6, all classes were identified as True Positives and True Negatives, except for class B, class J (Incorrect) and class O (Correct), which were classified as False Positives.

4.4. Outdoor Testing

Testing in this category was conducted five times outdoors with a testing distance of approximately ±70 cm and the camera height aligned with the tester's chest. The lighting level was around ±400 LUX, measured using the Lux Light Meter application. The author wore red clothing, a striped shirt, and maroon long sleeves and used a white background (fabric) for certain letters such as "Z." Accessories such as a wristwatch and ring were also worn. Meanwhile, the other actor wore black clothing with a similar white fabric background for specific letters like "Z." The testing results were recorded manually, letter by letter, from A to Z.

The following is an example of a test image using accessories with two hands in an outdoor setting.

Fig 18. Testing with accessories (a correct)

Fig 19. Testing with accessories (an incorrect)

The following is the confusion matrix for Test 1 using accessories in an outdoor setting, as shown in Table 4.

Table 4. Confusion Matrix IndorTesting (Correct)

Correct Class	Accessory-Based Outdoor Testing				Outdoor Testing with Striped Shirt				Different Actor Outdoor Testing				Long-Sleeve Outdoor Testing				Laptop Camera Outdoor Testing			
	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
A Correct	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0
B Correct	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0
C Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
D Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
E Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
F Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
G Correct	1	1	0	0	1	1	0	0	0	1	0	1	1	1	0	0	1	1	0	0
H Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
I Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
J Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0
K Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
L Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
M Correct	1	1	0	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	0	0
N Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
O Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0
P Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Q Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
R Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
S Correct	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0
T Correct	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	0	0
U Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
V Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
W Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
X Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0
Y Correct	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Z Correct	1	1	0	0	1	1	0	0	1	1	0	0	0	1	0	1	1	1	0	0
Total																				

Table 5. Confusion Matrix IndorTesting (Correct)

Correct Class	Accessory-Based Outdoor Testing				Outdoor Testing with Striped Shirt				Different Actor Outdoor Testing				Long-Sleeve Outdoor Testing				Laptop Camera Outdoor Testing			
	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN
A Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
B Incorrect	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
C Incorrect	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0	1	1	1	0
D Incorrect	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
E Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
F Incorrect	1	1	1	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
G Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0
H Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
I Incorrect	1	1	1	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
J Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
K Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
L Incorrect	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0

M Incorrect	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0
N Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0	1	1	0	0
O Incorrect	1	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0
P Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Q Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
R Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
S Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
T Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
U Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
V Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
W Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0
X Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Y Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Z Incorrect	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
Total																				

Based on Table 4 and Table 5 above, all classes are classified as True Positive and True Negative. In contrast, class F (Incorrect), class I (Incorrect), class L (Incorrect), and class O (Incorrect) are classified as False Positive. The following is an example of a test image using a striped shirt with two hands in an outdoor setting.

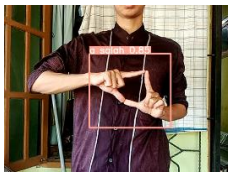
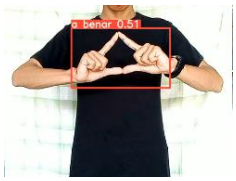


Fig 20. Striped shirttesting (a correct) **Fig 21.** Striped shirt testing (an incorrect)

Based on Table 4 and Table 5 above, all classes are classified as True Positive and True Negative. In contrast, class D (Incorrect), class F (Incorrect), class H (Incorrect), class M (Incorrect), and class S and T (Correct) are classified as False Positive. The following is an example of a test image with a different actor using two hands outdoors.



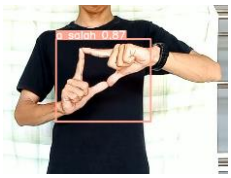


Fig 22. Testing with a different actor (a correct) **Fig 23.** Testing with a different actor (a incorrect)

Based on Table 4 and Table 5 above, 51 classes are classified as True Positive and 52 classes as True Negative, while class A (Correct), class B (Correct), class B (Incorrect), class C (Incorrect), and class M (Correct) are classified as False Positive, and class G (Correct) is classified as False Negative. The following is an example of a test image using long-sleeved clothing with two hands in an outdoor setting.

Fig 23. Long-sleeve testing (a correct) **Fig 24.** Long-sleeve testing (a incorrect)

Based on Table 4 and Table 5 above, 51 classes are classified as True Positive and 52 as True Negative. In contrast, class G (Incorrect), class M (Correct), class N (Incorrect), class S (Correct), and class T (Correct) are classified as False Positive, and class Z (Correct) is classified as False Negative. The following is an example of a test image using a laptop camera with two hands in an outdoor setting.

Fig 25. Laptop camera testing (a correct) **Fig 26.** Laptop camera testing (a incorrect)

Based on Table 4 and Table 5 above, all classes are classified as True Positive and True Negative. In contrast, class B (Incorrect), class C (Incorrect), class J (Correct), class O (Correct), class O (Incorrect), class W (Incorrect), and class X (Correct) are classified as False Positive.

4.5. Accuracy, Precision, and Recall Results

Based on the indoor and outdoor testing results, the average values of accuracy, precision, and recall were obtained and presented in tabular form.

Table 6. Evaluation results of indoor testing scenarios

No	Scenario	Accuracy	Precision	Recall
1.	Testing using accessories	0.954	0.912	1
2.	Testing with a striped shirt	0.889	0.831	0.947
3.	Testing with a different actor	0.948	0.901	1
4.	Testing with long sleeves	0.968	0.939	1
5.	Testing using a laptop camera	0.978	0.957	1

Table 7. Evaluation Results of Outdoor Testing Scenarios

No	Scenario	Accuracy	Precision	Recall
1.	Testing using accessories	0.957	0.917	1
2.	Testing with a striped shirt	0.933	0.875	1
3.	Testing with a different actor	0.956	0.928	0.987
4.	Testing with long sleeves	0.959	0.939	0.980
5.	Testing using a laptop camera	0.951	0.907	1

Note: Low : < 0.8 High : 0.8 – 0.9 Very High: ≥ 0.9

Based on Tables 12 and 13, several conclusions can be drawn from the indoor and outdoor testing as follows:

a. **Testing with accessories** (Table No. 1) showed high performance indoors and outdoors, with high accuracy, precision, and recall.
b. **Testing with a striped shirt** (Table No. 2) showed improved accuracy, precision, and recall performance from indoor to outdoor testing, with very high accuracy and high precision.
c. **Testing with a different actor** (Table No. 3) demonstrated exemplary performance in indoor and outdoor settings, with increased accuracy and high precision in outdoor testing.
d. **Testing with long sleeves** (Table No. 4) showed stable performance indoors and outdoors, with very high accuracy and precision.
e. **Testing using a laptop camera** (Table No. 5) demonstrated exemplary performance in indoor and outdoor settings, with a very high level of accuracy.

5. Conclusion

Using the pre-trained YOLOv5s variant as the object detection model is a suitable choice, as YOLOv5s offers relatively fast performance due to requiring fewer floating point operations and less memory than its larger variants. This makes the model more feasible to run on devices with limited memory capacity. By configuring the training process with a batch size of 16 and 120 epochs, YOLOv5s achieved excellent results. This is evidenced by the cross-validation results across iterations 1 to 10, showing an average precision of 0.990131 (or 99%) and an average recall of 0.999908 (or 99%). Indoor testing in each scenario yielded accuracy, precision, and recall values above 0.9, except in the scenario involving striped clothing, which recorded accuracy and precision values below 0.9.

Meanwhile, outdoor testing in each scenario also achieved accuracy, precision, and recall values above 0.9, except in the striped clothing scenario, where the precision value fell below 0.9. Based on the comparison between indoor and outdoor testing and the inclusion of accessories and various clothing types, it can be concluded that outdoor testing showed performance drops in specific scenarios, particularly in recall. However, overall, the system still demonstrated high accuracy and precision in outdoor testing.

References

[1] M. F. Nur Hayati Lilis, Nur Handayani Anik, Wahyu Sakti Gunawan Iriantoa, Rosa Andrie Asmara, Dolly Indra, "Classifying BISINDO Alphabet using Tensorflow Object Detection API," *Ilk. J. Ilm.*, vol. 15, no. 2, pp. 358–364, 2023.

[2] S. Anugerah, S. Ulfa, and A. Husna, "Pengembangan Video Pembelajaran Bahasa Isyarat Indonesia (BISINDO) Untuk Siswa Tunarungu Di Sekolah Dasar," *JINOTEP (Jurnal Inov. dan Teknol. Pembelajaran) Kaji. dan Ris. Dalam Teknol. Pembelajaran*, vol. 7, no. 2, pp. 76–85, 2020, doi: 10.17977/um031v7i22020p076.

[3] S. Daniels, N. Suciati, and C. Fathichah, "Indonesian Sign Language Recognition using YOLO Method," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1077, no. 1, p. 012029, 2021, doi: 10.1088/1757-899x/1077/1/012029.

[4] W. C. K. Sholikhatul Amaliya, Anik Nur Handayani, Muhammad Iqbal Akbar, Heru Wahyu Herwanto, Osamu Fukuda, "Study on Hand Keypoint Framework for Sign Language Recognition," *10.1109/ICEEIE52663.2021.9616851*, 2021, doi: 10.1109/ICEEIE52663.2021.9616851.

[5] G. B. Rosa Andrie Asmara, Muhammad Ridwan, "Haar Cascade and Convolutional Neural Network Face Detection in Client-Side for Cloud Computing Face Recognition," *2021 Int. Conf. Electr. Inf. Technol.*, 2021.

[6] M. C. Bagaskoro, F. Prasojo, A. N. Handayani, E. Hitipeuw, A. P. Wibawa, and Y. W. Liang, "Hand image reading approach method to Indonesian Language Signing System (SIBI) using neural network and multi layer perseptron," *Sci. Inf. Technol. Lett.*, vol. 4, no. 2, pp. 97–108, 2023, doi: 10.31763/sitech.v4i2.1362.

[7] E. Rahayu *et al.*, "LUMINA : Linguistic unified multimodal Indonesian natural audio-visual dataset," *Data Br.*, vol. 54, p. 110279, 2024, doi: 10.1016/j.dib.2024.110279.

[8] R. A. Asmara *et al.*, "YOLO-based object detection performance evaluation for automatic target aimbot in first-person shooter

games," *Bull. Electr. Eng. Informatics*, vol. 13, no. 4, pp. 2456–2470, 2024, doi: 10.11591/eei.v13i4.6895.
[9] R. Wulanningrum, A. N. Handayani, and A. P. Wibawa, "Perbandingan Instance Segmentation Image Pada YOLO8," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 11, no. 4, pp. 753–760, 2024, doi: 10.25126/jtiik.1148288.
[10] I. Zaeni, K. Kirana, Y. D. Mahandi, A. N. Handayani, and R. Fauzi, "Pelatihan SIBI (Sistem Isyarat Bahasa Indonesia) Berbasis Citra pada Siswa SLB Tunarungu Kota Malang," *J. Inov. Teknol. dan Edukasi Tek.*, vol. 1, no. 6, pp. 428–431, 2021, doi: 10.17977/um068v1i62021p428-431.
[11] Tazyeen Fathima, Ashif Alam, Ashish Gangwar, Dev Kumar Khetan, and Prof. Ramya K, "Real-Time Sign Language Recognition and Translation Using Deep Learning Techniques," *Int. Res. J. Adv. Eng. Hub*, vol. 2, no. 02, pp. 93–97, 2024, doi: 10.47392/irjaeh.2024.0018.
[12] M. Rivera-Acosta, J. M. Ruiz-Varela, S. Ortega-Cisneros, J. Rivera, R. Parra-Michel, and P. Mejia-Alvarez, "Spelling Correction Real-Time American Sign Language Alphabet Translation System Based on Yolo Network and LSTM," *Electron.*, vol. 10, no. 9, 2021, doi: 10.3390/electronics10091035.
[13] Z. S. Jannah and F. A. Sutanto, "Implementasi Algoritma YOLO (You Only Look Once) Untuk Deteksi Rias Adat Nusantara," *J. Ilm. Univ. Batanghari Jambi*, vol. 22, no. 3, p. 1490, 2022, doi: 10.33087/jiubj.v22i3.2421.
[14] R. A. Asmara, B. Syahputro, D. Supriyanto, and A. N. Handayani, "Prediction of Traffic Density using YOLO Object Detection and Implemented in Raspberry Pi 3b + and Intel NCS 2," *4th Int. Conf. Vocat. Educ. Training, ICOVET 2020*, pp. 391–395, 2020, doi: 10.1109/ICOVET50258.2020.9230145.
[15] D. O. Pratama, U. N. Malang, A. N. Handayani, and U. N. Malang, "Development of Embedded System Learning Module Using Project-based Learning Method for Industrial Electronics Department," *Lect. J. Pendidik.*, vol. 16, pp. 225–238, 2025.
[16] S. Tyagi, P. Upadhyay, I. Hoor Fatima, and A. Kumar Sharma, "American Sign Language Detection using YOLOv5 and YOLOv8," *Res. Sq.*, pp. 1–16, 2023, [Online]. Available: https://doi.org/10.21203/rs.3.rs-3126918/v1.
[17] D. Indra, R. Satra, H. Azis, A. R. Manga, and H. L, "Detection System of Strawberry Ripeness Using K-Means," *Ilk. J. Ilm.*, vol. 14, no. 1, pp. 25–31, 2022, doi: 10.33096/ilkom.v14i1.1054.25-31.
[18] M. A. A. K. Sanket Bankar, Tushar Kadam, Vedant Korhale, "Real Time Sign Language Recognition Using Deep Learning," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 9, pp. 193–199, 2023, doi: 10.22214/ijraset.2023.55621.
[19] M. Zaki, "Hand Keypoint-Based CNN for SIBI Sign Language Recognition," *Int. J. Robot. Control Syst.*, vol. 5, no. 2, pp. 813–829, 2025.
[20] D. Indra, S. Madenda, and E. P. Wibowo, "Feature Extraction of Bisindo Alphabets Using Chain Code Contour," *Int. J. Eng. Technol.*, vol. 9, no. 4, pp. 3415–3419, 2017, doi: 10.21817/ijet/2017/v9i4/170904142.
[21] P. Choirina and R. A. Asmara, "Deteksi Jenis Kelamin Berdasarkan Citra Wajah Jarak Jauh Dengan Metode Haar Cascade Classifier," *J. Inform. Polinema*, vol. 2, no. 4, p. 164, 2016, doi: 10.33795/jip.v2i4.77.
[22] N. H. Amir, C. Kusuma, and A. Luthfi, "Refining the Performance of Neural Networks with Simple Architectures for Indonesian Sign Language System (SIBI) Letter Recognition Using Keypoint Detection," *Ilk. J. Ilm.*, vol. 17, no. 1, pp. 64–73, 2025.
[23] R. A. Asmara, U. D. Rosiani, M. Mentari, A. R. Syulistyo, M. N. Shoumi, and M. Astiningrum, "An Experimental Study on Deep Learning Technique Implemented on Low Specification OpenMV Cam H7 Device," *Int. J. Informatics Vis.*, vol. 8, no. 2, pp. 1017–1029, 2024, doi: 10.62527/joiv.8.2.2299.
[24] Y. N. Faudah, I. D. Ubaidah, F. F. Ibrahim, Nur Taliningsih, N. K. Sy, and M. A. Pramuditho, "Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 3, p. 728, 2022, doi: 10.26760/elkomika.v10i3.728.
[25] A. T. Hermawan, I. A. E. Zaeni, A. P. Wibawa, Gunawan, W. H. Hendrawan, and Y. Kristian, "A Multi Representation Deep Learning Approach for Epileptic Seizure Detection," *J. Robot. Control*, vol. 5, no. 1, pp. 187–204, 2024, doi: 10.18196/jrc.v5i1.20870.
[26] C. Suardi, A. N. Handayani, R. A. Asmara, A. P. Wibawa, L. N. Hayati, and H. Azis, "Design of Sign Language Recognition Using E-CNN," *3rd 2021 East Indones. Conf. Comput. Inf. Technol. EIConCIT 2021*, pp. 166–170, 2021, doi: 10.1109/EIConCIT50028.2021.9431877.
[27] A. P. Wibawa and F. Kurniawan, "Enhancing Teks Summarization of Humorous Texts with Attention-Augmented LSTM and Discourse-Aware Decoding," *Int. J. Eng. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 156–168, 2025.
[28] R. Sutjiadi, S. Sendari, H. W. Herwanto, and Y. Kristian, "Generating High-quality Synthetic Mammogram Images Using Denoising Diffusion Probabilistic Models: A Novel Approach for Augmenting Deep Learning Datasets," *2024 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2024 - Proc.*, pp. 386–392, 2024, doi: 10.1109/ICITSI65188.2024.10929446.
[29] A. M. Sarosa, A. Badriyah, R. A. Asmara, M. K. Wardani, D. F. Al Riza, and Y. Mulyani, "Performance Analysis of MobileNET on Grape Leaf Disease Detection," *2024 Int. Conf. Adv. Inf. Sci. Dev. (ICAISD). IEEE*, vol. 64–68, 2024.
[30] M. O. Syahputra and L. Rosnita, "Analysis of Public Sentiment Toward Celebrity Endorsement On Media Social Using Support Vector Machine," *Int. J. Eng. Sci. Inf. Technol.*, vol. 4, no. 3, pp. 118–127, 2024.