Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2025. Vol. 4. No. 2; e-ISSN: 2808-4519

YOLOv8 Algorithm to Improve the Sign Language Letter Detection System Model

Madyanto[1*], Rudi Kurniawan[2], Yudhistira Arie Wijaya[3]

[1]STMIK IKMI Cirebon
madyanto96@gmail.com[1]*

Abstract

Sign language is a language used by individuals with hearing disabilities. Their limitations in socializing often require a tool to help them engage in social interactions. Therefore, by utilizing the YOLOv8 model, this research aims to improve the accuracy and efficiency of visual-based sign language detection, particularly for the alphabet of the Indonesian Sign Language System (SIBI). The goal of this study is to evaluate the performance of the YOLOv8 model in detecting sign language to identify the SIBI alphabet. The methodology includes training and testing the YOLOv8 model on a dataset consisting of videos and photos captured using a laptop camera. Each video is converted into images by extracting individual video frames, followed by preprocessing, data transformation, and data mining using YOLOv8 to generate bounding boxes, labels, and confidence scores for detected objects. With a complexity of 168 layers and more than 11 million parameters, this model is capable of consistently detecting sign language with an average inference speed of 4.6 ms per image. Detection results for each letter class show very high success rates, especially for letters such as D, F, N, O, and Q, which achieved accuracy of up to 96%. Overall, 22 out of 26 letters demonstrated "excellent" detection results (above 90%), while 4 letters (H, M, T, Z) showed "good" detection results (86%-89%).

Keywords: YOLOv8, Object Detection, Visual Gesture Recognition, Sign Language, Hearing Impairment

1. Introduction

Hand sign language is a common communication tool used by individuals with disabilities to interact with one another, particularly through hand gestures [1]. Hearing impairments pose significant challenges in communication with others. Deafness, or hearing loss, refers to the inability to hear in one or both ears and is often addressed with hearing aids or sign language. This form of language relies on hand and body movements as the primary medium of communication.

Visual sign language, such as hand gestures or facial expressions, has great potential in improving human interaction through computers, which can be applied in security systems, facial recognition software, and other automation applications. The implementation of an automatic sign language recognition system can help translate hand signals into text or letters from letters a to z. Through the application of the YOLOv8 algorithm in visual cue detection through laptop cameras, it aims to evaluate the performance of the model as well as explore its potential in sign language recognition as a communication aid through object detection. Although YOLOv8 is known to be fast, improvements in detection accuracy and speed are still needed, especially for real-time applications. The limitation of representative datasets is one of the obstacles in the development of an effective sign language recognition model.

Research on the development of a sign language recognition system that can process input from video data real-time using You Only Look Once (YOLO). This research continues the previous work by applying the model YOLOv8 to optimize visual cue detection and improve accuracy and speed in sign language recognition applications [2]. The importance of data augmentation in overcoming the limitations of the dataset, this technique increases data variation and helps the model learn better to detect the position of the hand before the classification of the hand shape is carried out [3][4].

The main objective of this study is to evaluate the performance of the YOLOv8 model in visual cue detection, specifically in the context of sign language, and to compare the effectiveness of this model in both local and cloud computing environments. The results of this study have several significant implications in the field of visual cue recognition and computer vision technology. First, improving the accuracy and speed of hand signal detection using the YOLOv8 model can facilitate the development of a more inclusive communication system, especially for sign language users. This has the potential to reduce communication barriers and expand the accessibility of technology for communities with special needs. Second, a comparison between the deployment of the model in on-premises and cloud computing environments provides important insights into the effectiveness and limitations of each platform, which can guide the development of computer vision-based applications in a variety of technical contexts. The implementation of YOLOv8 in a cloud environment, despite the challenges faced with camera access, shows that uploaded video-based solutions can be an effective alternative. In addition, the study helps in identifying areas that need further optimization, both in terms of processing speed and detection accuracy, so as to lead to the development of better models in the future.

The discovery also provides the basis for further research on the integration of signal detection technology in real-world applications, such as security systems, human-computer interaction, and assistive devices. Thus, the results of this research contribute to the advancement of object detection and sign language recognition technology, as well as paving the way for further innovation in this field.

2. Literature Review

Research on sign language recognition has grown rapidly with various approaches to computer vision-based methods and machine learning. In this literature review, several previous studies are presented that are relevant with a focus on sign language detection and recognition methods, including YOLO, CNN, and other learning algorithms.

The theoretical basis in this study is based on various journals that discuss sign language detection. The following is a summary of the main theories taken. CNNs are used in various studies to detect and recognize sign language in real-time, such as developing SIBI alphabet sign language recognition applications using CNNs [5]. CNNs are also used in recognizing numbers in SIBI [6], and for real-time sign language detection with TensorFlow Object Detection [7][8]. a combination of CNN and RNN to recognize the sign language alphabet in real-time. RNNs enable the processing of data that has temporal dependencies, such as hand gestures in sign language [1]. YOLOv8 algorithm to detect Indonesian sign language (BISINDO) in real-time. YOLO was chosen because of its speed and accuracy in detecting objects [9]. SIBI's sign language detection system uses the SSD method to detect the movement of the alphabet in real-time. This method is effective for the recognition of objects in images [10].

3. Research Methods

The YOLOv8 algorithm was chosen because of its speed and better accuracy in detecting objects. The data analysis used in this study is the Knowledge Discovery in Databases (KDD) method.

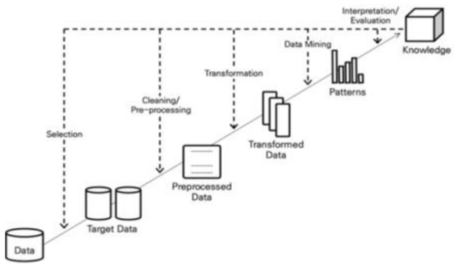


Fig. 1: Stages of Research Methods

The following are the stages of the research method using the Knowledge Discovery in Databases (KDD) Method along with its activities and descriptions as seen in Table 1 as follows:

Table 1: Stages of Research Methods

No	Stages	Activity	Activity Description
1	Selection	Data Selection	Select images that suit the research objectives and analysis needs.
2	Preprocessing	Data Cleaning	Clean and prepare the data for further analysis so that there is no duplication of data.
3	Transformation	Data Transformation	Performs labeling to convert images into text format so that they can be read by the YOLOv8 algorithm.
4	Data Mining	Proses Data Mining	Conducting YOLOv8 model training using processed data. to explore patterns or relationships contained in the data. In the epoch process used is 100 times.
5	Evaluation	Model Evaluation	Identify the extent to which the YOLOv8 algorithm model can correctly recognize objects and reduce detection errors.

Based on Table 1 of the research method, in this study, the Knowledge Discovery in Database (KDD) method is used for the data analysis technique. The following are the stages and processes that exist in Knowledge Discovery in Database:

1. Selection
   At the beginning of the stage there is a selection stage, where data selection is the process of analyzing relevant data from the database because it is often found that not all data is needed in the data mining process, the selection process is carried out on image object data.
2. Preprocessing
   The next stage is the preprocessing stage. At this stage, the stage of cropping and resizing the image is carried out for the purpose of YOLO input data.
3. Transformation
   At the stage of image data, it will be converted into data that is suitable for data processing that can be read by the YOLOv8 algorithm to convert data into data training, data testing and data validation.
4. Datamining

In the data mining stage, this research conducted training on the YOLOv8 model using processed datasets. During training, the model will learn to recognize patterns and features associated with traffic objects. Then to achieve a good balance between accuracy and training speed, 100 epochs were chosen for the data testing data training.

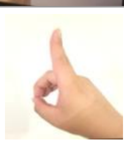
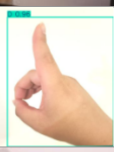
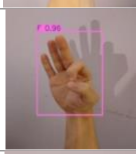
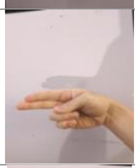
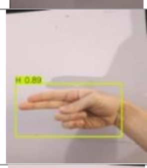
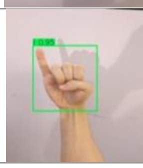
5. Evaluation

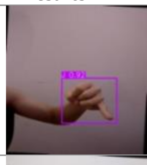
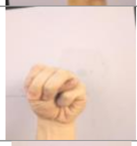
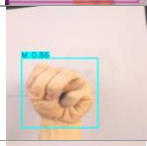
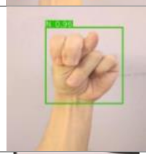
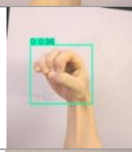
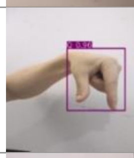
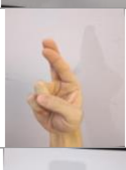
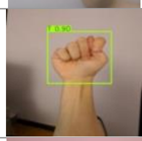
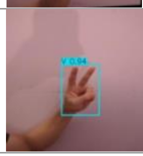
   This stage is carried out by the process of evaluation and interpretation of the rules that have been obtained in accordance with the predetermined objectives. At this stage, it applies a model that has been trained on a test dataset to perform object detection on traffic images that have never been seen before. Then use evaluation metrics such as Precision, Recall, and F1 Score to measure model performance. These metrics help identify the extent to which the model can correctly recognize objects and reduce false positives.

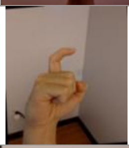
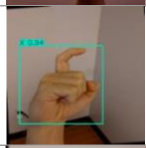
4. Results and Discussion

This study shows that the YOLOv8 model applied is able to achieve an impressive level of precision, recall, and F1 Score in sign language detection, with a precision value of 100%, recall of 99%, and F1 Score of 89%. The model test on 144 images with various sign variations showed an average mAP50 value of 0.96 and an mAP50-95 of 0.811, demonstrating the model's good ability to recognize signal patterns under various conditions that have been defined in the signal.yaml file.

Table 2: Test Dataset

No.	Classes	Test Data	Detection Results	Information
1.	A			The detection result is very good, which is 95%
2.	B			The detection result is very good, namely 94%
3.	C			The detection result was very good, namely 92%
4.	D			The detection result is very good, 96%
5.	And			The detection result is very good, which is 95%
6.	F			The detection result is very good, 96%
7.	G			The detection result is very good, which is 95%
8.	H			The detection result is quite good, which is 89%
9.	I			The detection result is very good, which is 95%

No.	Classes	Test Data	Detection Results	Information
10.	J			The detection result was very good, namely 92%
11.	K			The detection result was very good at 93%
12.	L			The detection result was very good, namely 91%
13.	M			The detection result is quite good, which is 86%
14.	N			The detection result is very good, 96%
15.	Or			The detection result is very good, 96%
16.	P			The detection result was very good, namely 91%
17.	Q			The detection result is very good, 96%
18.	R			The detection result is very good, namely 94%
19.	S			The detection result was very good at 93%
20.	T			The detection result is very good, namely 90%
21.	In the			The detection result is very good, namely 90%
22.	V			The detection result is very good, namely 94%

No.	Classes	Test Data	Detection Results	Information
23.	W			The detection result is very good, which is 95%
24.	X			The detection result is very good, namely 94%
25.	Y			The detection result was very good, namely 92%
26.	Z			The detection result is quite good, which is 88%

The results in table 2 of the test dataset are the results of testing the YOLOv8 algorithm where before conducting the test, the data mining process is carried out first. The following is a description of the datamining code to conduct training and testing 100 epochs:

```
from datetime import datetime
start = datetime.now()
results = model.train(data='isyarat.yaml', imgsz=640, epochs=100, batch=32, name="deteksi_isyarat")
end = datetime.now()
```

Fig. 2: Code Datamining

Once the code is executed, the YOLOv8 algorithm will be trained using the dataset that has been defined in the signal.yaml file. The algorithm will adjust the model's weights over 100 epochs, with an image size of 640 pixels and a batch of 32, to identify patterns in the sign language data. During the training, YOLOv8 will optimize the model parameters in an iterative manner to improve the accuracy of object detection (sign language) based on metrics such as precision, recall, and f1-score based on the following metrics:

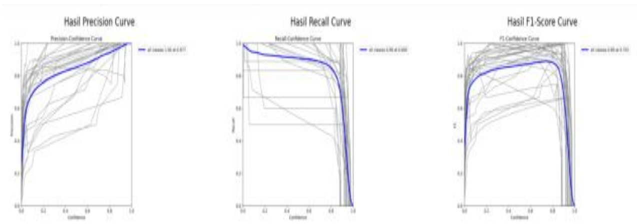


Fig. 3: Precision value, recall value and f1-score value

The results of the 100-epoch training show the progress of the model training on each epoch from 1 to 100. The number of instances varied during training, contributing to a better recognition of data variations. The detection performance for each class of alphabet signs (A–Z) varies, with some signs, most achieving accuracy levels above 90%, with the highest percentages for letters D, F, N, O, and Q reaching 96%. The model shows excellent performance in detecting most letters, but there are some letters such as H (89%), M (86%), and Z (88%) that have lower accuracy, suggesting that detection in these letters can be improved to achieve higher accuracy consistency.

To make an argument for this sign language detection, this code will be executed at runtime that will utilize the YOLOv8 model that has been trained to automatically detect objects on the input video. The following is the runtime code that is run to test the video to get different results before and after using the YOLOv8 algorithm.

```python
import os
from ultralytics import YOLO

# Load a pretrained model
model = YOLO("/content/ultralytics/runs/detect/deteksi_isyarat/weights/best.pt")

# Define path to video file
source = "/content/ultralytics/sampel_video.mp4"

# Define save directory
save_dir = "/content/ultralytics/runs/detect/predict"

# Create save directory if it doesn't exist
os.makedirs(save_dir, exist_ok=True)

# Check if the file exists
if os.path.exists(source):
    # Run inference on the source
    results = model.predict(source, save=True, save_dir=save_dir)  # Ensure results are saved to the specified directory

    # Optionally, process the results
    for result in results:
        print(result)

    print(f"Hasil disimpan di: {save_dir}")
else:
    print(f"File tidak ditemukan: {source}")
```

Fig. 4: Runtime Video Detection Code

Once the runtime code in figure 4 is executed, the YOLOv8 algorithm is loaded from the specified directory, which contains the best training weight. The video to be processed is also specified, along with a directory to store the detection results. If the storage directory doesn't already exist, the code automatically creates it. Next, the code checks for the presence of a video file if it is available, an inference process is executed to detect objects in the video, and the detection results are stored in a pre-prepared directory. The results of each detection are processed and displayed, along with a storage location notification. If the video is not found, the code will issue a warning.

Fig. 5. Before and after signal detection

Each detection comes with a label and a score. These labels provide information about the category of objects, while the score reflects the model's level of confidence in the accuracy of that detection. As such, the model shows great potential in real-time applications for automatic sign language detection, with consistent results in a variety of test situations.

5. Conclusion

The results show that the YOLOv8 model achieves a precision of 100%, a recall of 99%, and an F1 Score of 89%, with an average value of mAP50 of 0.96 and mAP50-95 of 0.811. In addition, an average inference time of 4.6 ms per image indicates the efficiency of the model. Thus, these results have a very positive impact on the sign language detection system. The detection results for each letter class showed a very high success rate, especially on letters such as D, F, N, O, and Q which achieved an accuracy of up to 96%. Overall, 22 out of 26 letters showed "very good" detection results (above 90%), while 4 letters (H, M, T, Z) had "quite good" detection results (86%-89%). This shows that the YOLOv8 model can consistently recognize different typefaces in sign language even under diverse image conditions, as has been tested on datasets with variations. Thus, the detection results, which show a high success rate, indicate that the YOLOv8 model can recognize various typefaces in sign language consistently and with high accuracy.

6. Suggestions

For further development, some suggestions can be considered i.e. to ensure that the model can be generalized well in various situations and backgrounds, it is recommended to expand the dataset with images that include more varied lighting conditions, viewing angles, and backgrounds. Given the potential of real-time applications, especially on mobile devices, it is necessary to optimize the model to make it lighter so that it can function smoothly on devices with limited power and processing capacity. To increase usefulness, the model can be expanded to include sign languages from different countries, so that it can support broader sign communication globally.

References

[1] D. Yolanda, K. Gunadi, and E. Setyati, "Pengenalan alfabet bahasa isyarat tangan secara real-time dengan menggunakan metode Convolutional Neural Network dan Recurrent Neural Network," *J. Infra*, 2020, [Online]. Available: http://publication.petra.ac.id/index.php/teknik-informatika/article/view/9791

[2] S. Daniels, N. Suciati, and C. Fathichah, "Indonesian sign language recognition using yolo method," *IOP Conf. Ser. …*, 2021, doi: 10.1088/1757-899X/1077/1/012029.

[3] R. I. Borman, B. Priyopradono, and A. R. Syah, "Klasifikasi Objek Kode Tangan pada Pengenalan Isyarat Alphabet Bahasa Isyarat Indonesia (BISINDO)," *Semin. Nas. Inform. dan Apl.*, no. September, pp. 1–4, 2018.

[4] U. Fadlilah, A. K. Mahamad, and B. Handaga, “The Development of Android for Indonesian Sign Language Using Tensorflow Lite and CNN: An Initial Study,” *J. Phys. Conf. Ser.*, vol. 1858, no. 1, 2021, doi: 10.1088/1742-6596/1858/1/012085.

[5] M. Sholawati, K. Auliasari, and ..., “Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (CNN),” *JATI (Jurnal Mhs. …*, 2022, [Online]. Available: https://ejournal.itn.ac.id/index.php/jati/article/view/4507

[6] M. B. S. Bakti and Y. M. Pranoto, “Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network,” *Pros. SEMNAS INOTEK …*, 2019, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/inotek/article/view/504

[7] A. H. Gustsa and G. S. Permadi, “Sistem Deteksi Bahasa Isyarat Secara Realtime Dengan Tensorflow Object Detection dan Python Menggunakan Metode Convolutional Neural Network,” *Inov. J. Ilm. Inov. …*, vol. 7, no. 2, pp. 1–10, 2023, [Online]. Available: https://ejournal.unhasy.ac.id/index.php/inovate/article/view/4116

[8] I. B. A. Peling, I. M. P. A. Ariawan, and G. B. Subiksa, “Deteksi Bahasa Isyarat Menggunakan Tensorflow Lite dan American Sign Language (ASL),” *J. Krisnadana*, vol. 3, no. 2, pp. 90–100, 2024, doi: 10.58982/krisnadana.v3i2.534.

[9] M. A. S. Andhika Bayu Pangestu, Muhamad Rafi Muttaqin, “SISTEM DETEKSI BAHASA ISYARAT INDONESIA (BISINDO),” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 5, pp. 9891–9897, 2024.

[10] S. Apendi, C. Setianingsih, and ..., “Deteksi Bahasa Isyarat Sistem Isyarat Bahasa Indonesia Menggunakan Metode Single Shot Multibox Detector,” *eProceedings …*, 2023, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19322