ANALYSIS OF DEEP LEARNING ACCURACY FOR BISINDO SIGN LANGUAGE TRANSLATION

RIZAL SENGKEY[1], BRAVE A. SUGIARSO[2], DIRKO G. S. RUINDUNGAN[3]
Dept. of Electrical Engineering, Sam Ratulangi University Manado
e-mail: rizal.sengkey@unsrat.ac.id[1], brave@unsrat.ac.id[2], dirko@unsrat.ac.id[3]

Abstract: Communication between the deaf and hearing people is still a challenge, especially due to the lack of understanding of the sign language they often used. To overcome this, the author developed an Indonesian Sign Language (BISINDO) translation and recognition application by utilizing the YOLOv4 deep learning model. This study aims to analyze the accuracy of the model in detecting and translating sign language in real-time. The research stage begins with creating a model. The model will be trained using the BISINDO dataset which can be accessed through an open-source dataset provider site for computer vision, and the performance evaluation will be carried out through several metrics such as Intersection over Union (IoU), Precision, Recall, and Mean Average Precision (mAP). The expected result is a model that is able to recognize and classify sign language with high accuracy based on the predetermined metrics. This research is useful in providing inclusive technological solutions to facilitate communication between deaf and hearing people, while increasing accessibility in the use of sign language.
Keywords: Accuracy; BISINDO; Deep learning; Sign language.

A. Introduction

Communication between Deaf individuals and the general public continues to face significant barriers due to the limited understanding of Indonesian Sign Language (BISINDO). Therefore, a technological solution capable of accurately and real-time translating hand gestures is needed. Previous studies have demonstrated that deep learning-based approaches, particularly YOLO models, show promising performance in object detection and gesture recognition tasks [1][2]. However, their application in the context of BISINDO remains limited and has not yet achieved optimal accuracy, efficiency, and generalization capability across various sign variations. This gap highlights the need for research focused on the development and performance analysis of YOLO models for BISINDO detection and translation, while also providing a novel contribution through a systematic evaluation of model accuracy using metrics such as Intersection over Union (IoU), Precision, Recall, and Mean Average Precision (mAP) [3]. By addressing this gap, the present study offers novelty in the form of real-time BISINDO translation system modeling and performance testing, as well as the provision of a dataset that can be utilized for future research in the field of computer vision. This study aims to develop and analyze the accuracy of a YOLO-based model for real-time BISINDO detection and translation as a step toward creating more inclusive technology, while also providing a practical solution to improve communication accessibility for the Deaf community in Indonesia.

B. Method

Machine Learning Model Development

The development of this artificial intelligence system utilizes a deep learning approach, which is a subset of machine learning. Deep learning enables computers to learn complex patterns directly from large amounts of data. This approach has been widely adopted in various fields, including computer vision, natural language processing, and speech recognition. One of its main advantages is the ability to automatically extract relevant features without extensive manual intervention. The effectiveness of a deep learning model often depends on the quality and quantity of the training data. Proper model tuning and evaluation are also essential to achieve optimal performance. The machine learning model development

 P-ISSN 2622-9110
E-ISSN 2654-8399

process consists of several stages, including dataset selection, dataset preparation, model design and training, and model evaluation.

During the data collection process, the author utilized a dataset containing labeled and annotated BISINDO sign language images. This dataset can be accessed through an open-source computer vision dataset repository. The dataset consists of two types of files: image files (.jpg) and their corresponding text files (.txt), both sharing the same filename. The text files contain annotation data associated with the respective images. The dataset includes various BISINDO hand gestures representing different letters. Each image was carefully reviewed to ensure that the annotations matched the corresponding sign language gesture. Data quality is an important factor in achieving reliable model performance. The collected images were stored and organized systematically to facilitate the training process. Furthermore, the use of labeled data helps the model learn the visual characteristics of each sign more effectively. In addition to the dataset, the author also collected supplementary image data using a camera to increase data diversity, which is expected to improve the model's accuracy and generalization performance.

The next stage in developing the YOLOv4 model involves the use of an annotated image dataset. Annotation is the process of labeling image data by assigning bounding boxes and corresponding object class names to objects within an image [4]. For the dataset obtained from the dataset provider, each image had already been annotated with the appropriate labels. However, the data collected by the author had not yet been annotated, requiring manual annotation prior to model training. In this study, annotations were applied to the hand regions forming the gestures in the images. The annotation process was carried out using YOLO Mark, a data annotation tool specifically designed for annotating images to be trained using the YOLOv4 algorithm.

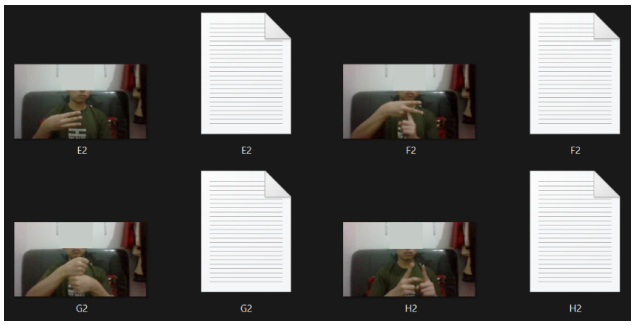

Figure 1. Annotation Process Using YOLO Mark

As shown in Figure 1, a bounding box is assigned by drawing a rectangle around the object and labeling it with the corresponding class name, for example, “f” to represent the sign language gesture for the letter F. This process is performed individually for each image. For every annotated image, a corresponding text file is generated containing the annotation data associated with that image.

The final step in preparing the dataset before it is used for model training is dataset splitting. For all image datasets and their corresponding annotation data, 60% of the data is allocated for training, 20% for validation, and the remaining 20% for testing.

In the development of this application, the object detection model used is YOLOv4. This model employs an architecture consisting of convolutional layers and several important hyperparameters that must be carefully configured.

These hyperparameters, along with all convolutional layer settings, are defined in a configuration file with the “.cfg” extension, which is loaded during the training, testing, and deployment phases of the machine learning model.

For the design of this model, the author configured the hyperparameters with the values presented in Table 1.

TABLE I
HYPERPARAMETER CONFIGURATION

Hyperparameter	Value
batch	64
subdivisions	16
width	416
height	416
channels	3
momentum	0.949
decay	0.0005
angle	0
saturation	1.5

In Table 1, the value of max_batches was determined using Equation (1).

$$max\_batches = number\ of\ classes \times 2000 \quad (1)$$

In Equation (1), the number of classes used is 27, consisting of 26 classes representing the letters A–Z and one class for person detection. However, the person class ultimately has minimal influence on the model's performance. Meanwhile, the step values used are 80% and 90% of the max_batches value, respectively.

In its development, the model is not trained from scratch using the sign language dataset. Instead, it is first trained using pre-trained weights, meaning that the model has previously been trained on the COCO dataset, which contains images and labels from 80 different object categories, in order to reduce training time [5]. The pre-trained weights are then combined with the weights obtained from training on the newly collected sign language dataset. By using a pre-trained model, the developed model is expected to achieve higher accuracy with a shorter training duration [6].

The testing process is conducted to ensure that the developed model is sufficiently accurate for implementation in the application system. Testing is performed using BISINDO data that is similar but not identical to the data used during training. The machine learning model used for this object detection testing stage is the same model that will be implemented in the system, specifically the model with the highest mAP value obtained during training [7]. The test result is considered correct if, after an image is input, a bounding box with the appropriate class label is successfully formed around the sign gesture in the image [8].

Accuracy Measurement Method

There are several interrelated metrics used to evaluate the performance accuracy of an object detection model, including Intersection over Union (IoU), Precision, Recall, and Mean Average Precision (mAP).

When two bounding boxes are present—namely the predicted bounding box from the deep learning model and the ground-truth bounding box from the labeled dataset—IoU is defined as the ratio between the area of intersection and the total area of union of the two bounding boxes [9]. Thus, IoU can be calculated using Equation (2).

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \quad (2)$$

P-ISSN 2622-9110
E-ISSN 2654-8399

IoU is used to determine whether a prediction is considered correct or incorrect by classifying each prediction result into three categories: True Positive (TP), False Positive (FP), or False Negative (FN). In the case of sign language detection, if the model successfully detects and correctly predicts the sign language letter, the prediction is categorized as a TP. If the model detects the object but produces an incorrect prediction, the result is categorized as an FP. If the model fails to detect an existing sign gesture and therefore provides no output or assumes that no gesture is present, the result is categorized as an FN [10].

Precision is a metric used to measure the accuracy of an object detection model's predictions. This value is obtained by calculating the ratio between the number of correct detections (TP) and the total number of detections, including both correct and incorrect ones [11]. This metric can be calculated using Equation (3).

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

A high precision value indicates that the model rarely produces incorrect predictions.

Recall measures how many of the actual existing letters are successfully detected by the model. The recall value is obtained by calculating the ratio between the number of correct detections (TP) and all possible positive instances, including both detected and undetected objects [12]. This metric can be calculated using Equation (4).

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

If the recall value is low, it indicates that many sign language letters are missed by the model and therefore fail to be detected. Conversely, a high recall value indicates that the model is able to detect almost all letters.

Mean Average Precision (mAP) is an evaluation metric used to measure the accuracy of an object detection model and is commonly used to select the best-performing weights for building the detection system. The mAP value is calculated by first computing the Average Precision (AP) for each class, ranging from letters A–Z, and then taking the mean of all AP values [13]. This metric can be expressed as shown in Equation (5).

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \quad (5)$$

A higher mAP value indicates that the model performs better in detecting and producing accurate predictions across all letters.

C. Results and Discussion

The methods used to analyze the accuracy of the BISINDO sign language detection model include Intersection over Union (IoU), Precision, Recall, and Mean Average Precision (mAP). These metrics provide a comprehensive evaluation of the model's detection performance. IoU measures the overlap between predicted and ground-truth bounding boxes. Precision evaluates the proportion of correct detections among all detections made by the model. Recall indicates the model's ability to identify all relevant sign language gestures. Meanwhile, mAP summarizes the overall detection performance across all classes. All of these evaluation metrics are automatically calculated by executing the following command in the Windows Command Prompt:

darknet.exe detector map data/obj.data cfg/yolov4-obj.cfg backup/yolov4-obj_best.weights -iou_thresh 0.5 [14]

The output of the above command is shown in Figure 2:

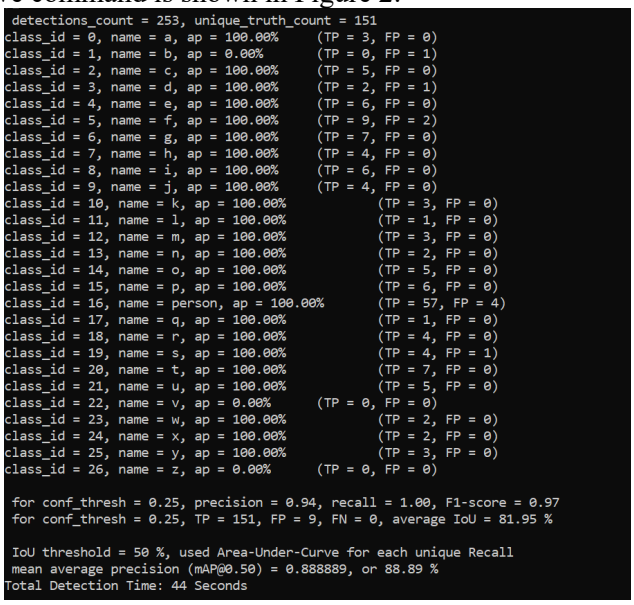

Figure 2. Output of mAP Calculation

D. Conclusion and Suggestions

This study successfully developed a deep learning model for BISINDO sign language detection covering letters A to Z using the YOLO object detection architecture. The developed detection model has previously been implemented in a web-based application, resulting in a system capable of translating sign language from Deaf individuals into text that can be understood by the general public. The application is able to detect sign language gestures for each letter in real time using a camera and display the detection results as translations. The resulting model demonstrates strong performance, as evidenced by machine learning evaluation metrics, including an Intersection over Union (IoU) of 81.95%, Precision of 94%, Recall of 100%, and Mean Average Precision (mAP) of 88.89%. These results indicate that, on average, the model performs well in detecting and recognizing all letters in BISINDO sign language. However, further improvements remain possible, particularly to enhance the model's accuracy beyond 90% by utilizing a larger and more diverse dataset [15].

Based on the final results of this study, there are still many opportunities for further development and future research. These include developing a more accurate and robust detection model by exploring alternative neural network architectures or utilizing larger and more diverse datasets, as well as extending the sign language translation system to cover a broader scope, such as sign language representing full words rather than only individual letters.

References

[1] E. Tikasni, E. Utami, and D. Ariatmanto, "Analisis Akurasi Object Detection Menggunakan Tensorflow Untuk Pengenalan Bahasa Isyarat Tangan Menggunakan Metode SSD," *JURNAL FASILKOM*, vol. 14, no. 2, pp. 385–393, Aug. 2024, doi: 10.37859/jf.v14i2.7512.

[2] A. Nugroho, R. Setiawan, A. Harris, and Beny, "Deteksi Bahasa Isyarat Bisindo Menggunakan Metode Machine Learning," *Jurnal PROCESSOR*, vol. 18, no. 2, pp. 152–158, Nov. 2023, doi: 10.33998/processor.2023.18.2.1380.

[3] S. N. Budiman, S. Lestanti, H. Yuana, and B. N. Awwalin, "SIBI (Sistem Bahasa Isyarat Indonesia) berbasis Machine Learning dan Computer Vision untuk Membantu Komunikasi Tuna Rungu dan Tuna Wicara," *Jurnal Teknologi dan Manajemen Informatika*, vol. 9, no. 2, pp. 119–128, Dec. 2023, doi: 10.26905/jtmi.v9i2.10993.

[4] J. Holdsworth and M. Scapicchio, "What is deep learning?," IBM. Accessed: Feb. 03, 2025. [Online]. Available: https://www.ibm.com/think/topics/deep-learning
[5] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign Language Transformers: Joint End-to-End Sign Language Recognition and Translation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle: IEEE, Jun. 2020, pp. 10020–10030. doi: 10.1109/CVPR42600.2020.01004.
[6] O. Koller, N. C. Camgoz, H. Ney, and R. Bowden, "Weakly Supervised Learning with Multi-Stream CNN-LSTM-HMMs to Discover Sequential Parallelism in Sign Language Videos," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 9, pp. 2306–2320, Sep. 2020, doi: 10.1109/TPAMI.2019.2911077.
[7] "What are convolutional neural networks?," IBM. Accessed: Feb. 03, 2025. [Online]. Available: https://www.ibm.com/think/topics/convolutional-neural-networks
[8] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020.
[9] S. Thakar, S. Shah, B. Shah, and A. V Nimkar, "Sign Language to Text Conversion in Real Time using Transfer Learning," in *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, Bangalore: IEEE, Oct. 2022, pp. 1–5. doi: 10.1109/GCAT55367.2022.9971953.
[10] M. R. Ningsih, Y. J. Nurriski, F. A. Z. Sanjani, M. F. Al Hakim, J. Unjung, and M. A. Muslim, "Sign Language Detection System Using YOLOv5 Algorithm to Promote Communication Equality People with Disabilities," *Scientific Journal of Informatics*, vol. 11, no. 2, pp. 549–558, Jun. 2024, doi: https://doi.org/10.15294/sji.v11i2.6007.
[11] H. Alzarooni, H. Al Ali, S. Fadaaq, and T. Bonny, "AI-Based Sign Language Interpreter (GestPret)," in *Proceedings of the 2023 7th International Conference on Advances in Artificial Intelligence*, New York, NY, USA: ACM, Oct. 2023, pp. 79–85. doi: 10.1145/3633598.3633612.
[12] S. Ennaama, H. Silkan, A. Bentajer, and A. Tahiri, "Enhanced Real-Time Object Detection using YOLOv7 and MobileNetv3," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 19181–19187, Feb. 2025, doi: 10.48084/etasr.8777.
[13] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," *Electronics (Basel)*, vol. 10, no. 3, pp. 279–306, Jan. 2021, doi: 10.3390/electronics10030279.
[14] L. H. He, Y. Z. Zhou, L. Liu, W. Cao, and J. H. Ma, "Research on object detection and recognition in remote sensing images based on YOLOv11," *Sci Rep*, vol. 15, no. 1, p. 14032, Apr. 2025, doi: 10.1038/s41598-025-96314-x.
[15] W. Ying *et al.*, "A Survey on Data-Centric AI: Tabular Learning from Reinforcement Learning and Generative AI Perspective," Feb. 2025.