Advancing Real-Time Sign Language Detection: Comparative Analysis of YOLOv8 Against YOLO v5, and Roboflow

Eirand Jan C. Barcelo
Department of Electronics Engineering, College of Engineering, Technological University of the Philippines
Manila, Philippines
eirandjan.barcelo@tup.edu.ph

Jashameel Faith D. Basa
Department of Electronics Engineering, College of Engineering, Technological University of the Philippines
Manila, Philippines
jashameelfaith.basa@tup.edu.ph

Rafael B. Dumguina
Department of Electronics Engineering, College of Engineering, Technological University of the Philippines
Manila, Philippines
rafael.dumaguina@tup.edu.ph

Ronald B. Laz
Department of Electronics Engineering ,College of Engineering, Technological University of the Philippines
Manila, Philippines
ronald.laz@tup.edu.ph

Abegail A. Lopez
Department of Electronics Engineering ,College of Engineering, Technological University of the Philippines
Manila, Philippines
abegail.lopez@tup.edu.ph

Camela Trisha J. Romen
Department of Electronics Engineering ,College of Engineering, Technological University of the Philippines
Manila, Philippines
camelatrisha.romen@tup.edu.ph

Jessica S. Velasco
Department of Electronics Engineering, College of Engineering, Technological University of the Philippines
Manila, Philippines
jessica.velasco@tup.edu.ph

Lean Karlo S. Tolentino
Department of Electronics Engineering, College of Engineering, Technological University of the Philippines
Manila, Philippines
leankarlo.tolentino@tup.edu.ph

Edgar A. Galido
Department of Electronics Engineering, College of Engineering, Technological University of the Philippines
Manila, Philippines
edgar.galido@tup.edu.ph

***Abstract*— This study presents a comparative analysis of real-time sign language detection using YOLOv8, YOLOv5, and the Roboflow model. The research focuses on evaluating these models based on their speed, accuracy, and overall performance in object identification applications. Convolutional Neural Network (CNN) models were trained and tested for recognizing the alphabet and single words in sign language. The results reveal that YOLOv8 exhibits remarkable precision, accuracy, and mean Average Precision (mAP) for alphabet detection and maintains excellent speed and accuracy for single words. Despite these strengths, the choice of model ultimately depends on the specific requirements of the application, such as the desired accuracy, real-time constraints, and available processing capacity.**

Keywords—Sign Language, Roboflow, YOLOv5, YOLOv8

I. INTRODUCTION (*HEADING 1*)

YOLO (You Only Look Once) is a widely recognized algorithm that excels in the real-time detection of objects within images. Over the years, various iterations of YOLO have been developed, each improving upon the last in terms of detection speed and reliability [1]. This algorithm employs convolutional neural networks (CNN) and operates via a single forward propagation through the neural network for object detection. Unlike two-stage object detectors, YOLO assesses the entire image in a single pass, enhancing both speed and efficiency.

As technology has progressed, object detection tools have seen substantial advancements. The latest iterations, such as YOLOv5 and YOLOv8, have introduced enhancements that offer sharper, faster, and more flexible object recognition capabilities.[2] The integration of these models with Roboflow, a platform that optimizes machine learning pipelines, further enhances their functionality. Roboflow's ability to manage and preprocess data effectively allows these YOLO models to operate with reduced computational load while achieving higher precision on benchmark datasets like MS COCO.

II. RELATED WORKS

A. Two-stage vs. Single-stage Object Detectors: A Detailed Examination

Two-stage detectors, like the R-CNN family, are renowned for their methodical approach to object detection. These models operate in two phases: the first phase generates proposals for object locations, and the second classifies these proposals into specific categories, assigning precise coordinates [3]. While this method achieves high accuracy, it is often criticized for its computational inefficiencies, primarily due to the initial proposal generation step.

In contrast, single-step detectors such as YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) adopt a more streamlined approach. These models eliminate the proposal generation phase entirely, predicting object classes and locations across the entire image in one sweep[4]. This simplification significantly accelerates the inference

process, making these models particularly well-suited for real-time applications. Initially, the speed increase of single-step detectors came at the expense of lower accuracy compared to two-stage models. However, ongoing improvements have begun to close this gap, balancing the demands of locating and classifying objects efficiently.

B. YOLO's Evolutionary Path: Bridging the Gap

The YOLO architecture has undergone significant evolution, from its first version, YOLOv1, to the more recent iterations like YOLOv5 and YOLOv8. Each version has built upon the last, enhancing speed, accuracy, and computational efficiency. These improvements include sophisticated architectural changes for feature extraction, advanced prediction techniques, and optimized loss functions[5]. Such innovations have incrementally bridged the performance gap between YOLO and more traditional two-stage detectors, cementing YOLO's reputation as a robust framework for object detection.

C. Comparative Performance Analysis: Yolo Versus Two-stage Detectors

Evaluating YOLOv5 and YOLOv8 against traditional two-stage detectors presents a nuanced view of their capabilities. While initially lauded for groundbreaking speeds that facilitate real-time detection, early YOLO models often lagged behind two-stage detectors in accuracy. However, subsequent iterations, notably YOLOv5 and beyond, have shown marked improvements in precision, challenging the supremacy of two-stage models on benchmark datasets like MS COCO. [6] Enhancements in network design, training methodologies, and performance optimization have been crucial to these advancements.

D. Integration of YOLO with Roboflow

The integration of YOLO models with Roboflow offers a transformative potential for real-time object detection systems. Roboflow's platform enhances the training of YOLO models by providing robust data management and augmentation tools, which ensure the models are trained on well-prepared datasets. [7] This synergy is particularly beneficial for applications like sign language detection, where the diversity and complexity of gestures demand highly accurate and responsive models.

E. YOLOv5 and YOLOv8: Advancement in Real-time Object Deitection

Recent developments in YOLO technology, particularly YOLOv5 and YOLOv8, have set new standards in the field of object detection. These models have been tailored to improve accuracy and reduce computational demands, making them ideal for embedding in systems that require real-time operation, such as dynamic sign language interpretation. [8] Their ability to process images swiftly and accurately ensures that they can be effectively used in systems designed to aid communication for the deaf and hard-of-hearing communities.

III. METHODOLOGY

A. Research Design

This study will utilize a descriptive quantitative approach to assess the effectiveness of the real-time web-based framework for sign language recognition and Baybayin translation using the YOLOv8 model. The primary data collection method will involve distributing questionnaires to respondents. The collected data will be used to evaluate the model's acceptability and accuracy. The questionnaire design will be based on parameters adapted from the International Organization for Standardization (ISO) 25010, ensuring thorough information collection aligned with the study's objectives[9]. The selection of respondents who evaluated the proposed system was done through purposive sampling, focusing on deaf-mute individuals affiliated with the Bigay Buhay Multipurpose Cooperative - Novaliches, Liliw, from which the sample was drawn.

B. Data Collection

The image data utilized in this study originates from an individual signer who is deaf-mute, and it is acquired through computer vision techniques. A setup consisting of a 1080P full-HD web camera and a laptop is employed to continuously capture hand gesture images depicting Filipino number signs, the alphabet, and common Filipino words. The capturing process is guided by a presentation tutorial sourced from the "Tinig" dictionary website. The resulting images undergo validation by an FSL trained signer to ensure accuracy and adherence to FSL standards.

C. Data Description

The dataset comprises hand gesture images of Filipino Sign Language (FSL) numbers, alphabet, and common Filipino words. The images are captured using a 1080P full-HD web camera and laptop setup. They depict the natural hand gestures of an individual signer who is deaf-mute. The capturing process is guided by a presentation tutorial from the "Tinig" dictionary website. Subsequently, the images are validated by an FSL trained signer to ensure accuracy

D. Architectural Design of YOLOs

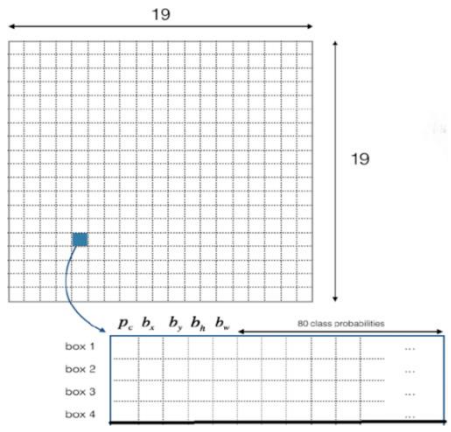
A convolutional neural network (CNN) can predict both the bounding boxes and the probabilities of different classes for all objects within an image. Named "You Only Look Once" (YOLO), this algorithm identifies objects and their positions in an image in a single pass. CNN excels at extracting features from visual input, efficiently transmitting low-level features from initial convolutional layers to deeper layers[10]. However, accurately identifying multiple objects and their precise positions within a single image poses a challenge. CNN addresses this challenge effectively through parameter sharing and the use of multiple filters, enabling robust object detection.

During the object detection process, the image or frame is divided into a grid of size $S \times S$. Each grid cell is responsible for predicting B bounding boxes, including their positions, dimensions, the probability of an object's presence in that grid cell, and conditional class probabilities. The key idea is that each grid cell aims to detect objects whose centers fall within its boundaries [11]. It accomplishes this by predicting suitable bounding boxes. Specifically, for each grid cell, the algorithm predicts a set of parameters for a

single bounding box. The first five parameters are specific to the bounding box, while the remaining parameters are shared across all bounding boxes within the grid, regardless of their number:

p_c	b_x	b_y	b_w	b_h	$p(c_1)$	$p(c_2)$	...	...	...	...	...	...	...	$p(c_n)$

where p_c represents probability of containing an object in the grid by the underlying bounding box, (b_x, b_y) indicate the center of the predicted bounding box, (b_h, b_w) represent predicted dimension of the bounding box, $p(c_i)$ means conditional class probability that the object belongs to i^{th} class for the given p_c and n is the number of classes/categories. A grid cell predicts $(B \times 5 + n)$ values, where B is the number of bounding boxes per grid cell. The output tensor shape would be $S \times S \times (B \times 5 + n)$ as we had divided the image into $S \times S$ grid cells. Figure 1 illustrates the final schematic of the output tensor prediction when the input image is divided into 19 × 19 grids as an example and four bounding boxes are predicted per grid wherein class probabilities are shared across all the bounding boxes for a specific grid. Confidence score (c_s) is computed for each bounding box per grid by multiplying p_c with Intersection over Union (IoU) between the *ground-truth* and *predicted-bounding-box*. If an object does not exist in the grid cell, confidence score would be zero. In the next step, we compute the class specific score (c_{ss}) for each bounding box of all the grid cells. This class specific score encodes both the probability of the class appearing in that box and how well the predicted box fits the object.



Generally, these bounding boxes differ in size, considering different shapes for capturing the different objects, known as anchor boxes. [12] An object in the image should be detected by a bounding box such that the center of the object should reside in that bounding box. However, there may be a possibility of residing centers of multiple objects in the same bounding box. Authors utilized a different term of anchor boxes to represent the bounding boxes corresponding to a single grid cell.

IV. Results and Discussion

A. Evaluation

The researchers have successfully trained several Convolutional Neural Network (CNN) models, including YOLOv8, YOLOv5, and the Roboflow model, to accurately classify Filipino Sign Language (FSL) input. After training the models, Google Colab and Roboflow provide the training graphs of each trained model. Listed below are the key parameters to note to comprehend the output graphs.

a) Mean Average Precison

mAP (mean Average Precision): A metric that averages precision across all classes, indicating overall detection accuracy.

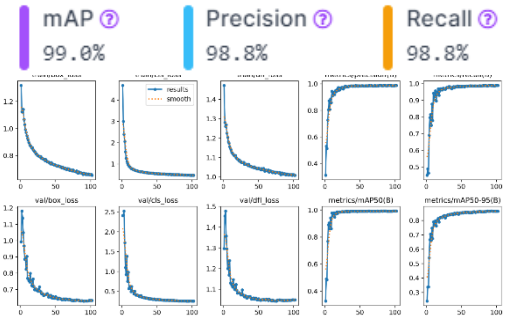


Figure 3 Single words, trained by Roboflow

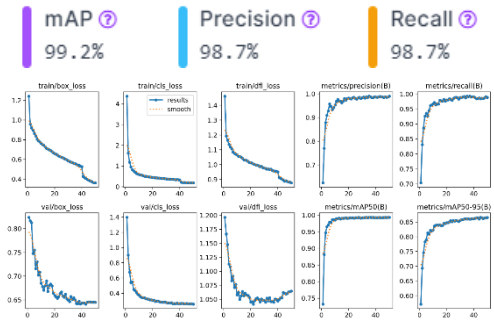


Figure 4 Single words, trained by YOLOv5

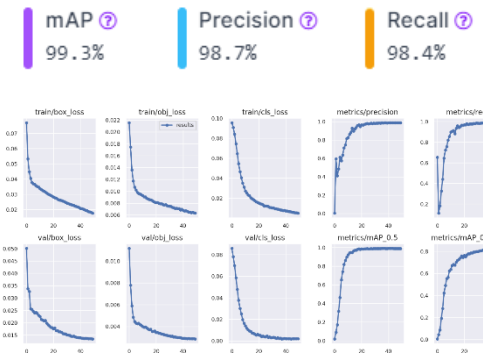


Figure 5 Single words, trained by YOLOv8

Model	maP
Roboflow-Single	99.2%
YOLOv5-Single	99%
YOLOv8-Single	99.3%

TABLE I. SUMMARY OF MEAN AVERAGE PRECISION

Table I summarizes the mean average precision (mAP) of various object detection models. Each model achieved a high mAP, all above 99%. The object detection models-Roboflow-Single, YOLOv5-Single, and YOLOv8-Single-achieved high mAP scores of 99.2%, 99%, and 99.3%

respectively. YOLOv8-Single showed the highest accuracy, indicating strong performance for tasks like sign language recognition and translation.

b) F1 Score

Evaluation metrics were employed to evaluate the performance of models in tasks like image classification and segmentation.

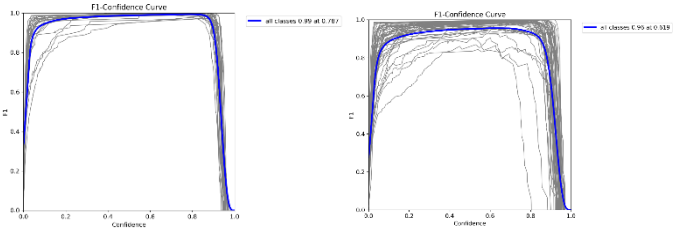


Figure 6 F1 curve of YOLOv5-Alphabet and Single Words

Figure 6 illustrates the performance of the model in two scenarios: the first graph shows a high overall F1 score of 0.99 at a confidence level of 0.787, demonstrating excellent precision and recall. The second graph displays an overall F1 score of 0.96 at a confidence level of 0.619 for single-word translations, indicating strong precision and recall at this threshold.

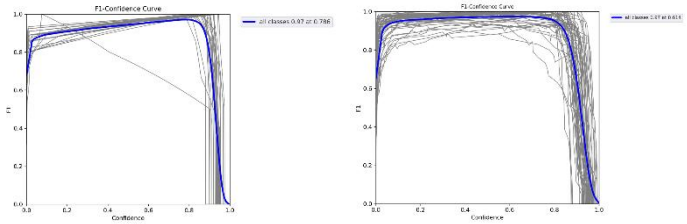


Figure 7. F1 curve of Roboflow-Alphabet and Single words

Figure 7 depicts two graphs: the first shows an overall F1 score of 0.97 at a confidence level of 0.786 for alphabet translation, highlighting excellent precision and recall. The second graph illustrates an overall F1 score of 0.97 at a confidence level of 0.614 for single-word translation, indicating strong precision and recall at this threshold. 4.17 shows a high overall F1 score of 0.97 at a confidence level of 0.786, demonstrating that the model maintains excellent precision and recall at this confidence threshold.

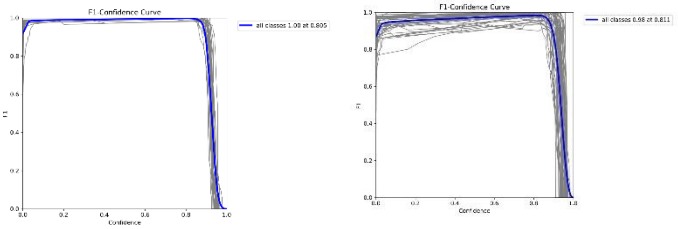


Figure 8. F1 curve of YOLOv8-Alphabet and Single words

Figure 8 first graph demonstrates perfect precision and recall with an overall F1 score of 1.00 at a confidence level of 0.805 for alphabet translation. The second graph shows very high precision and recall with an overall F1 score of 0.98 at a confidence level of 0.811 for single-word translation.

c) Accuracy testing using Bayesian

The researchers applied the Bayesian Theorem to evaluate the speed and accuracy of the trained CNN models, including YOLOv8, YOLOv5, and the Roboflow model. This method enabled a detailed comparison of each model's performance, highlighting their respective strengths and weaknesses in terms of processing speed and detection accuracy. The results provided a clear understanding of which model was most effective under different conditions, helping to determine the best model for real-time sign language recognition and translation tasks.

True Accuracy is found by using:
(P(C | A) + P(C | B)) / N
Where: N = the sample size

Model	True Accuracy
Roboflow	99%
YOLOv5	98%
YOLOv8	99.1%

TABLE II. ALPHABET ACCURACY TESTING USING BAYESIAN

Table II. above shows the three (3) machine learning models that exhibit excellent performance for alphabet letters. With accuracies of close to 99% for thirty (30) data inputs, it guarantees that the mAP, precision, and recall parameters from objective 2 are indeed true, proving its reliability.

Model	True Accuracy
Roboflow	92%
YOLOv5	91%
YOLOv8	91.7%

TABLE III. SINGLE WORDS TRANSLATION

Table III. demonstrates acceptable performance for static hand signs for single words. With accuracies of close to 90% for thirty (30) data inputs, it slightly deviates from the mAP, precision, and recall parameters from objective 2. But overall, accuracy is still competitive and acceptable as far as machine learning accuracy is concerned.

V. CONCLUSION

YOLOv8 emerges as the standout model among those tested, striking an impressive balance between speed and accuracy, achieving 93.1% accuracy in object identification applications. The machine learning model demonstrated excellent and satisfactory performance in both the alphabet and single-word tests. Additionally, the overall F1 score for the alphabet was 1.00 at a confidence level of 0.805, and 0.98 at a confidence level of 0.811 for single words. This indicates that the model achieves high precision and recall.

REFERENCES

[1] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. Available: https://doi.org/10.1109/CVPR.2016.91

[2] V. A. Kich et al., "Precision and Adaptability of YOLOv5 and YOLOv8 in Dynamic Robotic Environments," in Proceedings of the 11th IEEE International Conference on Cybernetics and Intelligent Systems (CIS), June 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2406.00315

[3] R. Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587. Available: https://doi.org/10.1109/CVPR.2014.81E. Lopez, "DECOLONISING MODERN LANGUAGES AND CULTURES," Blogs, https://blogs.ncl.ac.uk/decolonisesml/tag/baybayin/ (accessed Mar. 10, 2024).

[4] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Proceedings of the European conference on computer vision (ECCV), Springer, Cham, 2016, pp. 21–37. Available: https://doi.org/10.1007/978-3-319-46448-0_2

[5] V. A. Kich et al., "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," arXiv, 2023. Available: https://arxiv.org/abs/2304.00501

[6] R. Niloy, "Comparative Analysis Between YOLOv8 and Faster R-CNN," GitHub Repository. Available: https://github.com/R-Niloy/CPS843_Comparative-Analysis-Between-YOLOv8-and-Faster-R-CNN

[7] "American Sign Language Letters Object Detection Dataset - Roboflow," Roboflow. Available: https://public.roboflow.com/object-detection/american-sign-language-letters

[8] V. A. Kich et al., "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," arXiv, 2023. Available: https://arxiv.org/abs/2304.00501

[9] International Organization for Standardization, "ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models," 2011. Available: https://www.iso.org/standard/35733.html

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. Available: https://doi.org/10.1109/CVPR.2016.91

[11] J. Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. Available: https://doi.org/10.1109/CVPR.2016.91

[12] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517–6525. Available: https://doi.org/10.1109/CVPR.2017.690