Interpreting Sign Language Images into Text-Based using YOLOv7

Julius Sabilala[1], **Rv Jan Zaldo Bautista**[2], **Lea Mae Gambol Tortogo**[3], **Lerry Joy Ga Juntarciego**[4], **Charles Vincent Gayuma**[5], **Adrian Jaleco Forca**[6*]

[1,2,3,4,5,6*] College of Science and Technology, Guimaras State University-Mosqueda Campus, Alaguisoc, Jordan, Guimaras, 5044, Philippines

Abstract

Sign Language recognition remains a significant challenge due to variations in hand gestures, occlusions, and environmental factors. This study introduces a YOLOv7-based sign language recognition system designed to interpret American Sign Language (ASL) images into text in real time, enhancing communication accessibility for individuals with hearing and speech impairments. The primary objectives are to develop an accurate detection model, improve real-time gesture interpretation, and optimize system performance for accessibility. The study follows the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, including data collection, preprocessing, model training, evaluation, and deployment. The system was developed using YOLOv7, Python, and OpenCV, with an ASL dataset sourced from Roboflow and formatted in YOLOv7 PyTorch. Image preprocessing techniques such as normalization, resizing, and data augmentation were applied to enhance detection accuracy. The application integrates a real-time gesture recognition system, where detected ASL signs are instantly translated into text. Results demonstrate high detection accuracy for most ASL letters, but certain gestures such as J, Z, S, and T posed challenges due to similar hand shapes and motion-based characteristics. Optimization efforts included dataset expansion, refined annotations, and hyperparameter tuning to improve model precision. The system significantly enhances real-time ASL recognition, offering a scalable, AI-powered assistive tool for the deaf and hard of hearing community. This research contributes to machine learning-driven accessibility solutions, bridging the communication gap between sign language users and non-signers.

Keywords: *Assistive Technology; Machine Learning; Object Detection; Real-Time Translation; Sign Language Recognition; YOLOv7*

1. Introduction

For centuries, people who were hard of hearing or deaf have relied on communicating with others through visual cues. As deaf communities grew, people began to standardize signs, building a rich vocabulary and grammar that exists independently of any other language[1], [2]. According to the WHO (World Health Organization) report, over 466 million people are speech or hearing impaired, and 80% of them are semi-illiterate or illiterate [3]. Sign language is a means of communication through bodily movements, especially of the hands and arms, used when spoken communication is impossible or not desirable[4].

Sign language is based on hand gestures and used by those who are hearing- and/or verbally challenged for daily interactions and communication. The World Health Organization (WHO) has identified that more than 5% of the population of the world has hearing loss, including children. An estimated one out of ten people in the world will have a hearing disability by 2050 [5]. Sign language faces challenges like lack of

*Corresponding author. *E-mail address:* adrian.forca@gsu.edu.ph

Received: 26 May 2025, Accepted: 31 July 2025 and available online 31 July 2025
DOI: https://doi.org/10.33751/komputasi.v19i2.5260

standard grammar rules, absence of sign language dictionaries, and difficulty in translating spoken language [6]. A series of problems arise due to difficulty accessing information and joining discussions of communication support (such as interpreters or captions).

In the Philippines, 1.23% of the entire population is either deaf, mute, or hearing-impaired. As of 2009, the projected deaf population is already at 241,624 for those who are totally deaf, and 275,9J.2 for those who are partially deaf. That means that at least 517,536 people currently have very limited access to media and information because of their hearing impairment[7]. Machine Learning (ML) technologies have revolutionized various domains, including image processing and language interpretation. ML techniques facilitate the automatic recognition of sign gestures through computer vision, transforming visual inputs into text or speech. These advancements hold significant promise for enhancing communication accessibility for deaf individuals [8].

Communication is crucial, but for those with hearing impairments, sign language serves as an alternative. However, not everyone understands it. Deep learning, using TensorFlow Object Detection, SSD MobileNet V2, and CNN, offers a solution by training systems to recognize sign language patterns. A model trained on 489 images (381 for training, 98 for testing) was implemented in an Android app, achieving 50% accuracy in detecting sign language [9].

Recent studies have emphasized the incorporation of deep learning architectures, particularly Convolutional Neural Networks (CNNs) and object detection frameworks like YOLOv7, to improve the recognition accuracy of sign gestures. YOLOv7, known for its speed and efficiency in real-time object detection, can be employed to identify hand movements and positions accurately, significantly enhancing the processing of visual data [10]. This would foster greater inclusivity and accessibility for the Deaf community, thus improving their engagement with both digital and physical environments [11].

This study aims to design and implement a system that utilized technological advancements such as YOLOv7, CNNs, computer vision, machine learning algorithms, and natural language processing to accurately translate sign language gestures into text. By addressing the challenges associated with sign language recognition, the project sought to improve communication accessibility for hearing-impaired individuals and enhance social interactions between them and the hearing population. Utilizing image processing techniques within machine learning frameworks enabled the effective translation of sign language imagery into text. The use of this technology not only assisted in individual communication but also fostered a more inclusive environment by promoting understanding and integration among diverse groups.

2. Methods (10 pt, bold)

The cross-industry standard process for data mining, or CRISP-DM, served as the foundation for developing the Sign Language Interpreter study. This model provided a structured way to approach the machine learning workflow. It offered a structured, six-phase framework to guide the planning and execution of data science and machine learning. The process began with the Business Understanding phase and moved through subsequent phases as it developed, such as Data Understanding, Data Preparation, Modeling, Evaluation, before starting the last phase of Deployment[12] as shown in Figure 1.

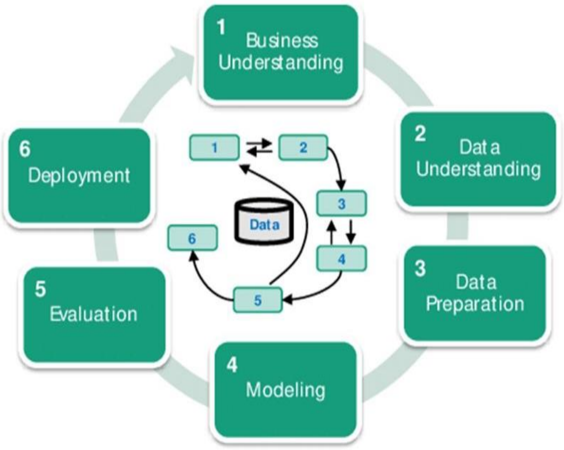


Figure 1. The cross-industry standard process for data mining (CRISP-DM)

2.1. CRISP-DM Phases

1. **Business Understanding Phase**. This is the first phase, which outlined the core project objective. Building a system to correctly interpret the sign language gesture from images and translate this into text for proper communication purposes between the hearing impaired and hearing population served as a great tool for people to communicate with one another. Interactions across all settings became more fluid because of it.

2. **Data Understanding Phase.** In this phase, relevant datasets, such as the ASL dataset, were gathered and analyzed to understand the nuances related to hand gestures, lighting conditions, and background noise that could impact model performance. The data had to be clean, diverse, and properly labeled to avoid biases and improve the model's ability to generalize well in real-world applications.

3. **Data Preparation phase.** During this phase, the received images were preprocessed to suit the training of the model. Image resizing, normalization, and data augmentation were some methods applied in order to increase the diversity of the dataset and enhance the robustness of the model. Additionally, label encoding was used to transform the sign language gestures into a form required by machine learning algorithms for proper processing. The received dataset was divided into three subsets: training, validation, and test data in order to predict appropriately.

4. **Modeling Phase.** In this phase, the YOLOv7 (You Only Look Once version 7) object detection framework was utilized. YOLOv7 represented a cutting-edge algorithm for real-time object detection, noted for its remarkable efficiency in identifying numerous objects within images and videos. This framework was particularly appropriate for the project as it possessed the capability to detect and accurately recognize multiple hand gestures in real time. YOLOv7 performed processing on the entire image in just one pass, which made it much faster than more traditional CNN-based approaches. This was particularly important for applications requiring real- time performance. The model was thus trained on this dataset; however, hyperparameters were tuned to improve performance in general. Several metrics, including mAP, precision, recall, and an F1-score, evaluated the model's performance and mainly focused on a reduction in the errors for detection.

5. **Evaluation Phase.** The trained YOLOv7 model was tested on a separate test dataset to ensure it had learned the capacity to generalize well to unseen data. This testing stage assessed the model's performance in the real world in terms of variability in lighting conditions, backgrounds, and hand shapes. For a system to be considered effective, it needed to be robust and have real-time recognition capability. If the model did not perform well, adjustments to the dataset or the model configuration were made to achieve higher accuracy and speed in detection.

6. **Deployment Phase.** In this phase, the project integrated YOLOv7 for real-time gesture detection, Flet for the frontend user interface, and Python for processing inputs. YOLOv7 detected ASL gestures from images or video feeds and translated them into text. The Flet interface enabled users to upload images or use webcams for gesture detection, displaying the corresponding text in real time. Python handled input processing, running the YOLOv7 model, and converting the detected gestures into text. This integration ensured a seamless, cross-platform application

2.2. Data Preparation

The dataset used for training the YOLOv7-based Sign Language Recognition system was sourced from Roboflow, a widely used platform for computer vision datasets[13], [14]. This dataset contains hand gesture images representing 26 American Sign Language (ASL) letters (A-Z).

2.2.1. Dataset Details

Format: YOLOv7 PyTorch-compatible dataset
Structure: Includes images and their corresponding annotation files inbounding box format
Categories: 26 classes (A-Z), each representing an ASL letter
Purpose: To train and evaluate the YOLOv7 model for real-time ASL recognition

To ensure consistency and enhance model performance, the dataset underwent preprocessing steps before training as shown in Table 1.

Table 1. Pre-Processing Steps

Pre-Processing Steps	Description
Image Resizing	All images were resized to 416x416 pixels, the recommended input size for YOLOv7[15]
Normalization	Image pixel values were normalized to improve learning efficiency
Augmentation	Techniques like rotation, flipping, and brightness adjustment were applied to enhance model robustness.
Noise Reduction	Blurry and low-quality images were removed to maintain dataset quality.

Each image was labeled with a corresponding .txt annotation file containing the bounding box coordinates in YOLO format.

YOLOv7 Label Format:

<class_id> <x_center> <y_center> <width> <height>

Example annotation for the letter "A":

0 0.5268420439811305 0.5070755689714356 0.7004771178074257 0.44822516231696535

Where:

0 → Class ID (ASL letter "A")
0.5268420439811305, 0.5070755689714356 → Bounding box center coordinates (normalized)
0.7004771178074257, 0.44822516231696535 → Bounding box width and height (normalized)

These labeled annotations allow the YOLOv7 model to accurately detect hand gestures during training.

2.2.2. Data Split for Training, Validation and Testing Sets

To ensure an effective training process, the dataset was divided into three subsets:

Table 2. Dataset Split and Its Purpose

Dataset Split	Percentage of Images	Number of Images	Purpose
Training Set	87.5% of dataset	1,512 images	Used to train the YOLOv7 model
Validation Set	8.33% of dataset	144 images	Helps tune hyperparameters and prevent overfitting
Test Set	4.17% of dataset	72 images	Used for final evaluation on unseen data

Table 2 presents the dataset split and its purpose: Training Set (87.5%) is used to train the YOLOv7 model, Validation Set (8.33%) helps fine-tune hyperparameters and prevent overfitting, and Test Set (4.17%) is reserved for final evaluation on unseen data to assess model generalization. This division ensures effective training, optimization, and performance assessment.

2.3. Modelling

The YOLOv7 Algorithm was selected due to its real-time object detection capabilities, efficiency, and accuracy. It is optimized for speed and performance, making it suitable for sign language recognition tasks[16]. The model was implemented using PyTorch and OpenCV, ensuring seamless integration with the application. YOLOv7 model was trained using the labeled American Sign Language (ASL) dataset from Roboflow. The training process involved, Loading the preprocessed dataset in YOLOv7 PyTorch format, Using GPU acceleration for efficient computation and Monitoring loss functions and performance metrics during training. Figure 2 shows the raw ASL letters samples from A-C from labeled dataset sourced from Roboflow.

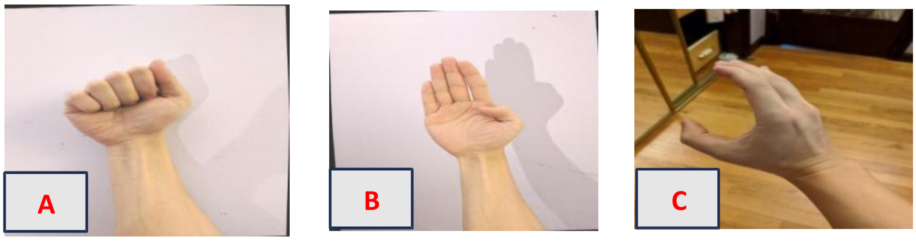


Figure 2. Sample ASL Letters (a) ASL Letter “A”, (b) ASL Letter “B” and (c) ASL Letter “C”

3. Results and Discussion

The researchers fine-tuned the hyperparameters of the YOLOv7 model to optimize its performance in recognizing American Sign Language (ASL) hand gestures. The tuning process involved adjusting parameters such as the learning rate, batch size, and confidence threshold to achieve a balance between accuracy and efficiency. To evaluate the impact of hyperparameter tuning, the model was tested before and after adjustments, and its performance was recorded based on precision, recall, and F1-score. In the implementation, The following parameters were adjusted: **Learning Rate** for Fine-tuned for stable convergence and **Batch Anchor Boxes** for Optimizing better object detection accuracy.

3.1. Model Accuracy

The prediction results of the trained YOLOv7 model was evaluated using various performance metrics to ensure accuracy and robustness. The model's reliability was assessed different conditions, such as varying lighting, hand orientations and backgrounds. The key evaluation metrics included classification accuracy, confusion matrix, precision, recall, and F1- score.

Classification Accuracy was one of the primary metrics used to measure the performance of the model as shown in Eq. (1). It was defined as the ratio of correctly predicted instances to the total number of instances in the dataset. Mathematically, it was represented as:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ number\ of\ Predictions\ made} \tag{1}$$

The Confusion Matrix was a crucial evaluation tool that provided a detailed breakdown of the model’s predictions by comparing the actual and predicted values. It was an N x N to identify the types of errors the model makes and Analyze misclassifications to refine the dataset. Figure 3 shows overall Confusion Matrix from A-Z presents a detailed evaluation of the YOLOv7 model's performance in recognizing ASL letters from A to Z. The confusion matrix visually represents the model’s predictions compared to actual labels, showing how often each letter is correctly classified and where misclassifications occur. It helps identify patterns of errors, such as confusion between similar hand signs, and provides insights for improving model accuracy through further training, data augmentation, or hyperparameter tuning.

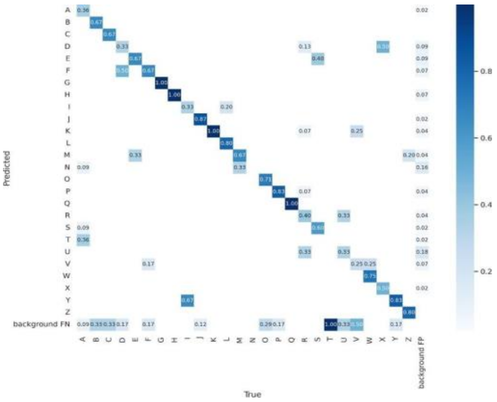


Figure 3. Overall Confusion Matrix from A-Z

As shown in Table 3, Per-Class Performance Metrics shows precision, recall, F1-score, and mAP for each ASL letter (A-Z), highlighting the model's accuracy and recognition challenges.

Table 3. Performance Metrics

Letter	Precision	Recall	F1-score	TP	FP	FN
A	1.0000	1.0000	1.0000	1	0	0
B	1.0000	1.0000	1.0000	1	0	0
C	1.0000	1.0000	1.0000	1	0	0
D	1.0000	1.0000	1.0000	1	0	0
E	1.0000	1.0000	1.0000	1	0	0
F	1.0000	1.0000	1.0000	1	0	0
G	1.0000	1.0000	1.0000	1	0	0
H	1.0000	1.0000	1.0000	1	0	0
I	1.0000	1.0000	1.0000	1	0	0
J	1.0000	1.0000	1.0000	1	0	0
K	1.0000	1.0000	1.0000	1	0	0
L	1.0000	1.0000	1.0000	1	0	0
M	1.0000	1.0000	1.0000	1	0	0
N	0.5000	1.0000	0.6667	1	1	0
O	1.0000	1.0000	1.0000	1	0	0
P	1.0000	1.0000	1.0000	1	0	0
Q	1.0000	1.0000	1.0000	1	0	0
R	1.0000	1.0000	1.0000	1	0	0
S	1.0000	1.0000	1.0000	1	0	0
T	0.0000	0.0000	0.0000	0	1	1
U	1.0000	1.0000	1.0000	1	0	0
V	1.0000	1.0000	1.0000	1	0	0
W	1.0000	1.0000	1.0000	1	0	0
X	1.0000	1.0000	1.0000	1	0	0
Y	1.0000	1.0000	1.0000	1	0	0
Z	1.0000	1.0000	1.0000	1	0	0

Table 4. Overall Model Performance

Metric	Value
Average Precision	0.9615
Average Recall	0.9615
Average F1-score	0.9490
Average Precision	0.9615
mAP (Mean Average Precision)	0.9490

In terms of overall performance of the model, As shown in Table 3, the model performs well on most classes, achieving perfect scores for the majority of sign letters. However, the letter 'N' has a precision of 50%, indicating some misclassification, while the letter 'T' has an F1-score of 0.0000, meaning the model failed to correctly identify this sign. To improve performance, additional training samples should be added for underperforming classes ('N' and 'T'), and YOLOv7 hyperparameters should be fine-tuned.

Figure 4 reveals that, real-time inference, the system successfully detects a sample hand gesture corresponding to the letter "A" in American Sign Language (ASL). The detected gesture is enclosed in a green bounding box, with the label "A" and a confidence score of 0.41 displayed above it. The system processes the video feed in real time, identifying and classifying the hand gesture based on the trained YOLOv7 model. The text "Press 'q' to quit" is also visible, indicating that the application is running interactively, allowing users to exit the inference mode when needed.

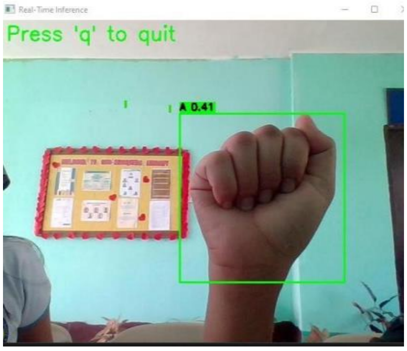
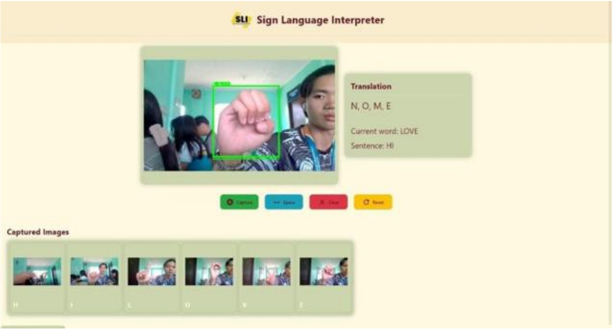


Figure 4. Real-Time Inference of Sign Language Detection

The trained YOLOv7 model was integrated into a cross-platform application built using Flet and Python. OpenCV was used for real-time image processing, and the model was embedded to enable live sign language interpretation.

The researchers tested the performance of the Sign Language Interpreter system by evaluating its accuracy and response time in detecting and interpreting sign language gestures. The system was tested in real-time using different hand gestures under varying lighting conditions and backgrounds to ensure its robustness and efficiency. The testing was conducted at different time frames throughout the day to assess its consistency in recognition performance.



Translation

N, O, M, E

Current word: LOVE

Sentence: HI

Figure 5. Sign Language Interpreter System Interface and Real-Time Translation Sign ***Language***

4. Conclusion

The study’s results demonstrated that the Interpreting Sign Language Images into Text- Based Using Machine Learning successfully achieved its objectives, providing real-time detection with high accuracy. This system helps students, teachers, and professionals by converting sign language gestures into text in real time. It uses machine learning to make communication easier for people who don’t know sign language, allowing them to interact better with the deaf and mute community. It can be used in schools, workplaces, and public places to improve understanding and inclusivity.

In conclusion, the Interpreting Sign Language Images into Text-Based Using Machine Learning serves as an innovative solution for enhancing communication accessibility. This system utilizes machine learning techniques to accurately recognize and convert sign language gestures into text, making interactions more efficient and inclusive for individuals who are unfamiliar with sign language. By automating the interpretation process, it helps bridge communication gaps, particularly in educational institutions, workplaces, and public services. Overall, the system provides a promising tool for fostering inclusivity and improving accessibility for the deaf and mute community

5. Recommendation

Future researchers can enhance the Interpreting Sign Language Images into Text Using Machine

Learning system by implementing several key improvements. Expanding the dataset with more diverse sign language images under various lighting conditions and hand orientations can improve model accuracy and robustness. Enhancing the system with both on-screen text display and voice output will make communication more accessible. Focusing on sign languages that use one sign per word can simplify interpretation and improve accuracy. Utilizing Google Colab for model training will leverage cloud-based GPU resources for faster and more efficient training. Implementing a training pause and resume function will allow researchers to stop and continue training without losing progress. Additionally, adding a zoom function to the live camera feed will enhance hand detection accuracy by reducing background noise

6. Acknowledgement

The author acknowledges Roboflow for providing a free dataset for the researchers delving in the field of Machine Learning. This significant platform provides a holistic access to more accessible dataset for image processing and machine learning model processing.

References

[1] D. Lillo-Martin, "Sign Language: Syntax," *Encyclopedia of Language & Linguistics*, pp. 351–353, Jan. 2006, doi: 10.1016/B0-08-044854-2/00241-8.

[2] J. Strickland, "How Sign Language Works." Accessed: May 24, 2025. [Online]. Available: https://people.howstuffworks.com/sign-language.htm

[3] "Deafness and hearing loss." Accessed: May 24, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[4] J. E. Pyers, "Sign Languages," Encyclopedia of Human Behavior: Second Edition, pp. 425–434, Jan. 2012, doi: 10.1016/B978-0-12-375000-6.00327-X.

[5] A. D. Goenawan and S. Hartati, "The Comparison of K-Nearest Neighbors and Random Forest Algorithm to Recognize Indonesian Sign Language in a Real-Time," Scientific Journal of Informatics, vol. 11, no. 1, pp. 237–244, Feb. 2024, doi: 10.15294/SJI.V11I1.48475.

[6] R. Attar, V. Goyal, and L. Goyal, "Development of Airport Terminology based Synthetic Animated Indian Sign Language Dictionary," Journal of Scientific Research, vol. 66, pp. 88–94, Jan. 2022, doi: 10.37398/JSR.2022.660512.

[7] L. K. S. Tolentino, R. O. S. Juan, A. C. Thio-ac, M. A. B. Pamahoy, J. R. R. Forteza, and X. J. O. Garcia, "Static Sign Language Recognition Using Deep Learning," Int J Mach Learn Comput, vol. 9, no. 6, pp. 821–827, Dec. 2019, doi: 10.18178/ijmlc.2019.9.6.879.

[8] A. Srivastava, R. Maity, A. Srivastava, and R. Maity, "Assessing the Potential of AI–ML in Urban Climate Change Adaptation and Sustainable Development," Sustainability 2023, Vol. 15, Page 16461, vol. 15, no. 23, p. 16461, Nov. 2023, doi: 10.3390/SU152316461.

[9] O. Mailani, I. Nuraeni, S. A. Syakila, J. Lazuardi, and P. I. Komunikasi, "Bahasa Sebagai Alat Komunikasi Dalam Kehidupan Manusia," Kampret Journal, vol. 1, no. 2, pp. 1–10, Jan. 2022, doi: 10.35335/KAMPRET.V1I1.8.

[10] P. J. C. Mendoza, A. F. Salo, N. D. Santiago, K. M. S. Mauricio, and A. G. C. Cano, "Sign Language Text Translator Using YOLOV7 Algorithm," Lecture Notes in Networks and Systems, vol. 1012 LNNS, pp. 433–444, 2024, doi: 10.1007/978-981-97-3556-3_35.

[11] P. A. Rodríguez-Correa, A. Valencia-Arias, O. N. Patiño-Toro, Y. Oblitas Díaz, and R. Teodori De la Puente, "Benefits and development of assistive technologies for Deaf people's communication: A systematic review," Front Educ (Lausanne), vol. 8, p. 1121597, Apr. 2023, doi: 10.3389/FEDUC.2023.1121597/BIBTEX.

[12] C. Schröer, F. Kruse, and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," Procedia Comput Sci, vol. 181, pp. 526–534, Jan. 2021, doi: 10.1016/J.PROCS.2021.01.199.

[13] "American Sign Language Letters Object Detection Dataset - v1." Accessed: May 26, 2025.

[Online]. Available: https://public.roboflow.com/object-detection/american-sign-language-letters/1

[14] "Roboflow Universe: Computer Vision Datasets." Accessed: May 26, 2025. [Online]. Available: https://universe.roboflow.com/

[15] J. Yap and T. Wiradinata, Safety Helmet Detection Based on YOLOv7 With Super-Resolution Reconstruction. 2024. doi: 10.1109/ICTIIA61827.2024.10761195.

[16] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," pp. 7464–7475, Aug. 2023, doi: 10.1109/CVPR52729.2023.00721.