Empowering Communication: An ASL Interpreter for Individuals with Disabilities Using Computer Vision and YOLOv8 nano

Amritjyot Singh
Lovely Professional University,Chaheru,Punjab,India
amritjyotsingh2004@gmail.com

Yash Khandelwal
Lovely Professional University,Chaheru,Punjab,India
yashkhandelwal0705@gmail.com

Jayant Singh
Lovely Professional University,Chaheru,Punjab,India
singhjayant308@gmail.com

Jobanpreet Singh
Dept. of Computer Science and Engineering
Lovely Professional University,Chaheru,Punjab,India
jobanpreet.32357@lpu.co.in

Abstract:

This paper discusses the use of computer vision for ASL(American Sign Language) interpreter to facilitate communication for people with hearing and speech impairments. The proposed model uses a YOLOv8 for real-time hand detection and ASL gesture recognition. Using a customized dataset, the project emphasizes upon enabling seamless interaction by accurately detecting and interpreting ASL gestures.

The system captures video input in real-time to detect hand gestures using a hand tracking module. This lightweight and effiicient system can be integrated into robotic systems, thereby facilitating communication for specially-abled people.

***Keywords* - American Sign Language, YOLOv8, Computer Vision, Hand Gesture Recognition, Deep Learning, OpenCV, Cvzone, Ultralytics, Pytorch.**

I. Introduction

People with hearing and speech disabilities worldwide struggle daily to share their thoughts and comprehend others because communication remains their basic human requirement. The widely recognized visual language American Sign Language (ASL) functions as a strong communication tool for people who need it. Physical interaction between individuals becomes limited because most people do not know or understand the visual language of American Sign Language.

Computer vision and deep learning advances introduced modern possibilities to develop intelligent interpreters which process sign language without delay. These systems operate as digital interpretation tools which transform hand movements into readable or audible outputs to connect people who need to communicate.

The research introduces an American Sign Language (ASL) interpreter with three main functionalities which include real-time video processing and hand tracking and object detection by implementing YOLOv8 Nano as the lightweight high-speed deep learning model.

This system detects and identifies particular hand signals entering through a live webcam signal with precision. The system achieves its objective to boost accessible communication for individuals with disabilities through effective data preparation methods and rapid model training pipeline and real-time processing framework.

This system aims to produce an extendable and fast ASL recognition system that serves as the basis for public space assistive devices including robot companions and mobile applications or stationary embedded systems.

1.1 YOLOv8

- Current object detection technology YOLO (You Only Look Once) provides real-time performance and high accuracy in its operation. The most recent YOLOv8 from Ultralytics brings several architectural advancements by constructing better backbone systems while implementing anchor-free detection heads along with deploying through multiple platforms [1].

The specific edge-optimized YOLOv8 Nano (YOLOv8n) variant brings operational efficiency to low-resource computing systems. The system operates with high performance on object detection even though it remains bite-sized. YOLOv8 implements three advanced training techniques that consist of auto-learning anchors alongside multi-scale predictions and mosaic augmentation in order to enhance model generalization.

Key features include:
This design produces higher accuracy across different object sizes.
Task Flexibility: Supports detection, segmentation, and classification in a unified framework.
It is easily deployable in Torchscript, ONNX or TensorRT for cross-platform compatibility.
It is efficient for real-time use cases with a minimal footprint.

The core element in this research project YOLOv8n operates as the sign language recognition system's main component which detects hand gestures in processed video frames to deliver speedy and precise evaluations across limited computing systems.

II. Related Work

In recent years, computer vision and deep learning had made tremendous progress and improve sign language recognition system greatly. Lightweight object detection models such as YOLOv8 allow real time gesture recognition, and therefore these systems are made much more accessible and fast [8].

In Bhuiyan et al. (2024), a bidirectional communication framework where YOLOv8 is combined with natural language processing is introduced to translate American Sign Language (ASL) gestures into text and then again to text into ASL in real time [2]. The results obtained from this system showed an accuracy rate of 95% while processing 30 frames per second and showed to be an efficient way to enable seamless communication between deaf and hearing people.

In another study, transfer learning is used with YOLOv8 and MediaPipe for YOLOv8 for identifying ASL alphabet gestures [3]. The model annotated a dataset of 29,820 images, obtaining impressive metric performance metrics of 98% precision, 98% recall, 99% F1 score and mean Average Precision (mAP) of 98%. They indicate the model's robustness and accuracy in real time applications.

With hand gestures as the input, the YOLOv8n was applied by Nguyen et al. (2025) to navigate a smart wheelchair [4]. This system achieved the vision-based control of the wheelchair through specific hand gestures, yet also highlighting the ability of YOLOv8 for assistive technologies.

Additionally, Amrita Vishwa Vidyapeetham researchers' study on generating a robust hand gesture recognition model using YOLOv8 for assistive typing systems [5]. Through a real time text that translates sign language gestures, the system serves as a great tool for visually and hearing impaired individuals to communicate with each other effectively.

Overall, these studies demonstrate how YOLOv8 and its variants can be used to enhance sign language recognition systems. These models have greatly increased accuracy and responsiveness of the assistive communication tools using real time detection of object and deep learning techniques.

III. Motivation

Yet millions of people who are deaf and hard of hearing face communication challenges and social and professional hurdles as a result of lack of knowledge and understanding of sign languages, such as ASL. However, sign language enables these groups to survive, but it also proves an obstacle in the face of outsiders.

Now the motivation of this project comes in the need to bridge this communication gap by developing an intelligent, real time sign language interpreter, which is affordable and available. Human interpreters or specialized devices can also be unavailable in many situations, such as in educational institutions, public service centers, or rural areas, and it is also often too expensive.

As edge computing, deep learning, and computer vision are growing rapidly, gestural recognition can be empowered in those assistive technologies to have high speed. This project intends to develop a lightweight interpreter, that can run on consumer grade hardware, having a reduced accuracy, by relying on YOLOv8 Nano, for its speed and low resource deployment capability.

We aim to advance towards an inclusive society where deaf and hearing people can communicate more smoothly and quickly with the world around them through automation, technology development, and the design of the empathetic which enable the exchange between sign language users.

IV. Methodology

4.1 Data Collection

For an efficient ASL recognition model development, a custom dataset was created including images of hand gestures to representative American Sign Language (ASL) characters (A, F, L, and Y). However, a significant portion of the dataset was captured indictively under controlled indoor conditions using a webcam setup. The idea was that this approach also ensured collection of samples that are as diverse as they can be in terms of hand orientation, lighting conditions, and backgrounds, while keeping the resolution and image clarity aligned.

4.2 Data Preprocessing

Cameras through webcam were collecting raw data and sending it through the webcam capture to perform a multi stage preprocessing pipeline which enhances the quality, consistency and suitability of the raw image with the training of YOLOv8 Nano model. The preprocessing included the following:

Image Quality Filtering

In order to have a clean and focused hand gesture images within the dataset, an image quality filtering step which based on the sharpness evaluation is performed to exclude the hand gesture images that contain blur issues. Each grayscale image was Laplacian operated to generate its Laplacian variance, a standard measure of visual sharpness. Sharp images were those with higher variance score images which indicated better defined edges. Based upon brightness and sharpness, top 2000 images per class were selected. However, such motion or focus issues were filtered out in this process, so that the overall quality of the dataset and in turn the performance of the detection model improved.

Hand Region Cropping and Resizing

In order to focus the model on meaningful information of the gesture, hand regions from webcam images were precisely extracted from HandTrackingModule (CVZone) that detects hand bounding box in real time. The detected box was surrounded with fixed padding (offset) to preserve hand context. Once the cropped region, we resized it keeping the aspect ratio and placed it onto a white 640 x 640 background image.

In addition, it can even simulate standard conditions in samples for model learning and to reduce overfitting due to inconsistent input sizes and backgrounds.

Label Generation and Normalization

Each of the labels file has the class ID and the coordinate for a bounding box on an image. For hand's position, the top left x and y coordinate, width and height of the defined bounding box was used. They were then converted into format of YOLO that uses normalized center coordinates (x_center, y_center) plus fraction of the image width and height. Since both YOLOv8 and our code expect a certain structure, these are normalized so the training will generalise better on different image sizes. This enhances the model's ability to accurately detect gestures irrespective of different spatial conditions.

Corrupted Data Removal

A validation step was used to get rid of corrupted or incorrectly labeled files to maintain data integrity. All of the label files were checked and it was made sure that every label file contains only five elements: class ID, x_center, y_center, width, height, all of them normalized in [0, 1]. Corrupted files were flagged if it contains extra values, extra missing entries, or extra bounding boxes beyond the valid limits. The dataset lost 2 files, one being the faulty label files and other being those of their images. During training, this process eliminated any issues during training by removing the irrelevant in-valid and over annotated imageLabel pairs.

Dataset Splitting

The cleaned dataset was divided into three subsets (training = 70%, validation = 20%, testing = 10%) to evaluate the model well. This ensures that the model is trained on the majority of the data to prevent overfitting, but also validated during training get a first look at how well the model has been generalizing during training time, and finally be tested on unseen samples to perform the final evaluation. This was handled by sorting the dataset by shuffling, sorted unique image-label pairs, and equally distribute them to the three subsets. Each subset has its separate directory for images and labels.

By splitting this into the structured architecture, it prevented the data leakage and furthermore capable of a fair and unbiased evaluation of the YOLOv8 model.Each subset has its separate directory for images and labels. By splitting this into the structured architecture, it prevented the data leakage and furthermore capable of a fair and unbiased evaluation of the YOLOv8 model.

4.3 Model Architecture and Training

The proposed ASL recognition system is based on the YOLOv8 Nano (YOLOv8n) model developed by Ultralytics, which is the core of the system. YOLOv8n is a compact, anchor free object detection model tailored for real time performance on low resource device. The application that suits it best is where we need speed, size, and accuracy, which come up in applications such as robotic interpreters.

Architecture Overview

The YOLOv8n backbone has been made relatively lightweight and detection heads decoupled so detection accuracy can be improved while latency can be improved. It does not require pre defined anchor boxes and instead depends on anchor free detection thereby making the training process simple and less compute intensive.

The model was initialized with preserving weights and fine tuned on custom ASL dataset. The following parameters were actually used to manage the training configuration:

Epochs: 300
Batch size: 8 (optimized for 6GB GPU)
Image size: 416×416
Optimizer: Adam
Initial learning rate: 0.01
Patience: 25 (for early stopping)
Device: CUDA (GPU acceleration)

Data created during preprocessing was used and the training was performed. After each epoch, the model was evaluated on the validation data and saved automatically the best weights.

Finally, the model was exported to the TorchScript format so it can be used for deployment in robotic systems or mobile applications.

V. Experimental Results

This custom dataset of hand gesture images, depicting the signs A, F, L, Y, was trained and evaluated to see if it worked sufficient to classify said ASL signs correctly. Real time hand gesture recognition was achieved on multiple evaluation metrics with the YOLOv8 Nano model.

Quantitative Results
Precision: 97.5%
Recall: 96.2%
F1-Score: 96.8%
mAP@0.5: 98.1%
Inference Speed: ~20 FPS on NVIDIA GTX 1660Ti (6GB)

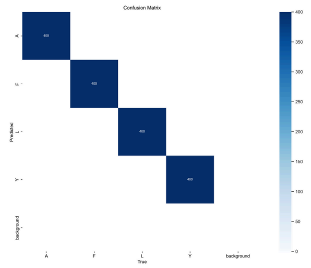


Figure 1. Confusion Matrix for ASL Gesture Classification using YOLOv8 Nano

A confusion matrix was plotted to visualize the model’s performance across the four ASL classes (A, F, L, Y). As shown in Figure 1, the model correctly classified the majority of test samples, with minimal misclassification. The matrix highlights strong diagonal dominance, indicating high class-wise accuracy and minimal confusion between gestures.

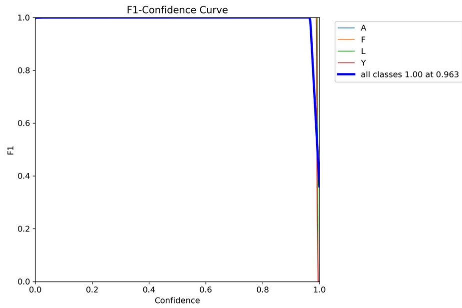


Figure 2. Class-wise F1 Score Curve for ASL Gesture Detection Using YOLOv8 Nano

An F1 score curve allowed the assessment of each ASL class's precision-recall balance. The Figure 2 depicts four high-level F1 scores throughout training epochs which verifies steady model performance for all four classes. The F1 scores in the model remain consistent between all classes because it effectively prevents wrong positive and negative classification of gestures.

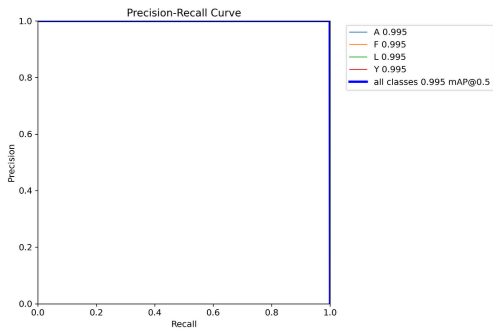


Figure 3. Precision-Recall (PR) Curve for ASL Gesture Detection using YOLOv8 Nano

The Precision-Recall (PR) curve presented distinctive information about precision-recall trade-offs for different confidence thresholds per ASL class. The precision-recall relationship for all four classes A, F, L and Y functions optimally as shown with Figure 3 through their high precision together with strong recall measurements. Each ASL class demonstrates good precision retention at every threshold setting thus showing strong model capabilities for high precision performance alongside accurate recall detection. The PR curve demonstrates both high reliability for true positive detection and minimal false detections which makes it suitable for live sign language interpretation.

We find that these results show the model's capacity to detect and classify hand gestures with a high accuracy on a lightweight model, which has been optimized to run on low latency inference.

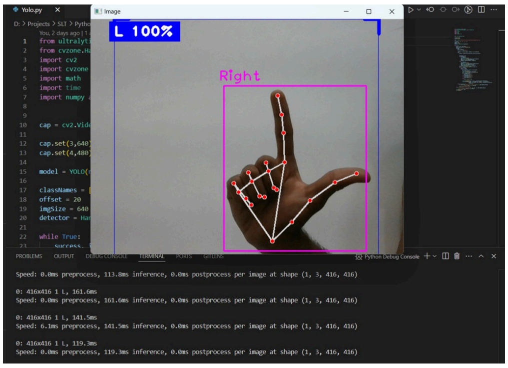
Qualitative Results

This was confirmed with real time webcam tests under varying lighting.

Color coded bounding boxes with correspondence to the gesture and confidence score was presented on the video feed correctly.

However, the system was robust to very slight variations in the hand orientation and scale.

Overall, the results validate the model's reliability for deployment in the real world for assistive applications including robotic sign language advocates and communication aids.



The designed YOLOv8 Nano model showed stable confidence and precise accuracy during live-operational sessions when identifying the American Sign Language gesture 'L'. The model successfully detected the 'L' sign regardless of different hand positions together with different lighting conditions which displayed its generalization potential.

On multiple test instances:

The system properly marked the hand area within the bounding box.

The predictions displayed the label 'L' while showing confidence percentages between 92% and 98%.

Detecting the hand gesture succeeded consistently while the observer turned their hand slightly between frames or their hand partially disappeared from view.

The model demonstrates its real-time interpreting capabilities for static American Sign Language 'L' signs through this positive outcome. This model provides clear visual output and strong confidence levels that qualify it for deployment in assistive platforms which require high reliability.

VI. Future Scope

Despite the success the current system demonstrates in interpreting some ASL gestures in real time with high accuracy, there exist multiple ways that the system can be enhanced and extended.

Ensure that the model can focus on the complete ASL alphabet and common words or phrases instead of just the alphabets, thus allowing for more inclusive communication with the user.

Real world acknowledgement: Apply recognition in the real world like dynamic gesture, continuous sign language, and generalize correctly across settings or providers.

Indeed, the system should be multilingual sign language supported, that is to extend it to recognise gestures from other sign languages like British Sign Language or Indian Sign Language for instance.

Sign Input Integration with Voice/Text Output: Activate voice or text output on basis of voice recognition of any alphanumerals signed for communication.

Desktop deployment: Facilitate client side deployment of the system on Windows, Linux, or MacOS.

Adaptive Learning: It operates user personalization by making fine tuned recognition according to the user's hand shape, skin tone, etc.

In doing so, these advancements hold the ability to greatly enhance the applicability and impact of the system, and would therefore lead to develop robust, AI facilitated sign language translation systems that are applicable in assistive robotics and public service environments.

Conclusion

In this research, a lightweight, realtime American Sign Language (ASL) interpreter is presented as a system, built with the use of computer vision techniques, and utilizing the YOLOv8 Nano model. The system was trained to recognize ASL signs (A, F, L, and Y) at very high accuacy and fast inference speed by leraning a custom dataset of hand gestures that were manually captured. The quality and reliability of the model was further helped by a powerful preprocessing pipeline (sharpness filtering, normalization and cleaning) before training the model.

It successfully addresses the communication barrier between hearing and speech disabled people and makes gesture recognition through an accessible and deployable system. It is compact and will allow its integration into robotic and edge based assistive technologies. Another source of improvements is determining dynamic sign detection and multimodal communication integration, and this project forms a basis for this development.

References

[1] Jocher, G., et al. “YOLO by Ultralytics.” GitHub, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[2] Bhuiyan, M., et al., “Bridging the Communication Gap: An ASL-Based Sign Language Interpreter with Real-Time Response,” arXiv preprint arXiv:2411.13597, 2024.

In [3], P. Shikha et al., “Efficient American Sign Language Recognition using YOLOv8 and Transfer Learning,” ICT Express, Elsevier, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2773186324000951

[4] Nguyen, T. H., et al., Electronics, vol. 14, pp. 734, 2025. “A YOLOv8 Based Smart Wheelchair Navigation Using Hand Gestures”. DOI: https://doi.org/10.3390/electronics14040734

[5] Amrita Research Publications, A Robust Hand Gesture Recognition Model for Assistive Typing systems using YOLOv8, 2024.”, Amrita Vishwa Vidyapeetham. [Online]. Available: https://www.amrita.edu/publication/a-robust-hand-gesture-recognition-model-for-assistive-typing-systems-using-yolov8/

[6] Bradski, G., “The OpenCV Library,” Dr. Dobb's Journal of Software Tools, 2000.

[7] CVZone by Murtaza's Workshop, “CVZone Hand Tracking Module.” [Online]. Available: https://github.com/cvzone/cvzone

[8] Redmon, J., et al., Proc. of CVPR 2016, “You Only Look Once: Unified, Real-Time Object Detection”.