Two-way Sign Language Translation

Mahmoud Y. Shams[1*], Alaa Hussien[1*], Hagar Ashraf[1*], Nada Nasser[2], Mariam Gabr[3], Hosam Mohamed[4], Atef Yasser[5]

[1,2,3,4,5]Faculty of Artificial Intelligence, Kafrelsheikh University, Kafrelsheikh, 33516, Egypt.

*Corresponding author(s). E-mail(s): mahmoud.yasin@ai.kfs.edu.eg; alaa.ai_0267@ai.kfs.edu.eg; hagar.ai_0473@ai.kfs.edu.eg;
Contributing authors: nada.ai_0462@ai.kfs.edu.eg; mariam.ai_0434@ai.kfs.edu.eg; hossam.ai_0296@ai.kfs.edu.eg; atef.ai_0332@ai.kfs.edu.eg;

Abstract

People with speech impairments or hearing loss frequently utilize sign language as a way of communication. Yet, the language is not understood by everyone. Communication between the deaf and the general public will be facilitated by the automatic translation of sign language into the alphabet or text. This paper proposes a mobile application that enables two-way sign language translation for seamless communication. We leverage deep learning, specifically YOLO v8 and YOLO v5 models, for accurate sign language recognition and translation. Our results indicate that YOLO v8 achieves high precision, recall, and mean average precision (MAP) scores of 98%, 97%, and 99%, respectively, on the Sign Hands dataset, and 95%, 96%, and 98%, respectively, on the American-sign-language dataset. Similarly, YOLO v5 demonstrates competitive performance with precision, recall, and MAP scores of 96.5%, 96.7%, and 98%, respectively, on the Sign Hands dataset, and 94%, 95%, and 98%, respectively, on the American-sign-language dataset. The system offers two key functionalities: 1) Text/Speech to ASL, which converts spoken or written English to text and then translates it into ASL animations using a 3D avatar, and 2) ASL to Text/Speech, which recognizes ASL gestures captured through the camera and translates them into text or spoken English. This mobile application has the potential to improve accessibility for both deaf and hearing communities.

Keywords: American Sign Language, YOLO v8, YOLO v5, Blender, Avatar, 3D Modelling, Mobile application, Android.

1 Introduction

Over 1.5 billion individuals worldwide suffer from hearing loss. In addition, the incorrect use of earbuds and headphones puts over one billion youngsters at danger of hearing loss. These conditions negatively impact children's development, schooling, and mental and physical health. Children who suffer from hearing loss may experience communication challenges and delayed language development. Regrettably, hearing loss is frequently not sufficiently accommodated in public and private contexts, which has a detrimental effect on academic performance and career prospects.

Sign language is very important these days. Humans can communicate nonverbally by using hand gestures to visually represent information to another person. This is known as sign language[1]. Individuals who are deaf or have trouble speaking often utilize sign language. Most everyday individuals do not acquire sign language skills because performing it is not necessary. However, individuals cannot just be able to interpret sign language instantly without having to go through a learning process if there is ever a need to do so. It is not possible for all deaf people to utilize the same sign language. The detection of sign language presents several difficulties because of its distinct features and the difficulty of capturing and deciphering sign language. Even though there are some obvious parallels, sign language is used differently in every nation. Communication between the deaf and the mute will be made easier with a system that can automatically recognize and interpret signs from pictures or videos to letters or text. For a reliable and timely operation in vision-based Sign Language Recognition (SLR), deep learning, a subset of machine learning, has gained popularity recently. Deep learning has become very important and reliable in sign language studies due to its outstanding performance in picture classification, object recognition, image capture, semantic segmentation, and human pose prediction.

Another benefit of deep learning approaches is their capacity to analyze big datasets, which is crucial for image processing. By comparing the pertinent hand model with the entire image that was taken and processed. Deep learning in vision-based SLR aims to produce an accurate classification. There is a noticeable trend in many institutions where American Sign Language (ASL)[2] is prioritized as it is suitable for training the models. "You Only Look Once" (YOLO) is a deep learning technique that has gained popularity recently for object detection. Convolutional neural networks (CNNs) are the foundation upon which YOLO was constructed, and it provides quick and accurate object recognition. In this research, we use Yolov5 and Yolov8 to recognize American Sign Language in real life. The overall functionality of our proposed system is demonstrated in Fig. 1.

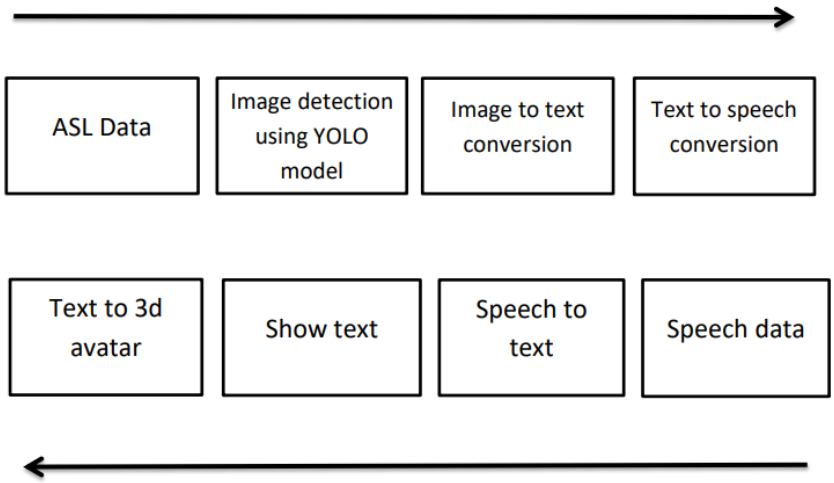


Fig. 1 Two-way communication between deaf and hearing people

2 Related work

Sign languages are characterized as a systematic set of hand movements with distinct meanings used by individuals with hearing impairments to communicate in daily situations. They communicate by moving their hands, faces, and bodies since they are visual languages. Globally, there are more than 300 distinct sign languages in use. Even though there are numerous sign languages, only a small fraction of people are familiar with them, which makes it challenging for those with special needs to interact with everyone informally. SLR offers a way to communicate without being familiar with sign language. It interprets a gesture into a language that is frequently spoken, like English. Numerous studies are being conducted to translate American Sign Language to text. One notable approach utilizes MediaPipe, an open-source framework for building multi-modal ML pipelines. Notably, MediaPipe offers the SSD-MobileNet-V2 [3] model, a lightweight and efficient detector for real-time applications.

2.1 Mediapipe

Gesture-based communication acknowledgment can advance the circumstance of a huge number of crippled individuals while managing ordinary people be that as it may, the utilization of communication via gestures for correspondence is restricted. Therefore, It is necessary to develop a more beneficial methodology so that those with hearing impairments can learn and improve their lives. The research of gesture recognition has made substantial use of conventional techniques such as skeleton tracking, different color glove-based monitoring, and body component tracking. Among other techniques, this problem has been solved with CNN, SVM, and deep learning. AI calculations have been produced for sign Language acknowledgment to make computer-based intelligence-based applications. Out of them, Machine learning

pipelines for time-series data, including audio, video, and other formats, are constructed using the MediaPipe[4] framework. It was first made available by Google on YouTube for real-time audio and video analysis. Researchers and developers can now incorporate and use this framework in their projects thanks to MediaPipe's public release in 2019.

In contrast to most high machine learning frameworks that require a lot of processing power, MediaPipe can function effectively. We can provide a model or computation for the application with the help of the MediaPipe system, and then aid the program by providing results that can be replicated at different stages. The MediaPipe system is made out of three significant parts: (1) execution assessment, (2) a system for gathering information from the sensor (3) a get-together of reusable parts. A chart comprising all the parts called the mini-computers is known as a pipeline, wherein each number cruncher is associated by channels through which the information streams (Fig. 2). Designers can make their necessary application by eliminating or depicting client-characterized number crunchers placed in the chart. The consequence of adding machines and channels makes an information stream graph. Hand signal acknowledgment with the MediaPipe structure is a trustworthy furthermore, the high-loyalty hand and finger-global positioning framework.

We can detect and track the movements and positions of both hands (Fig. 3) in real-time thanks to the Mediapipe module's reliable and effective hand pose estimate. We extract 21 landmarks total for each hand from the hand tracking module, recording their movements and spatial arrangement. These landmarks are crucial components of the sign language recognition model's later stages.

Mediapipe hands utilize a coordinated ML line of a few models cooperating : (1) A palm recognizer processes the caught hand picture, (2) A hand milestone model accepts the handled picture as info and returns the hand with 3D central issues as a result. (3) A motion acknowledgment model that processes the 3D hand fundamental concerns and classifies them into distinct motion arrangements. The recognition of the palm model results in an unequivocally edited image of the palm that is then shipped off the milestone model. This technique gets rid of information expansion, which is utilized in profound models to turn, scale, and flip pictures. The method of distinguishing hands is tedious what's more, troublesome because it includes collaborating with various sizes of hands, thresholding, and picture handling.

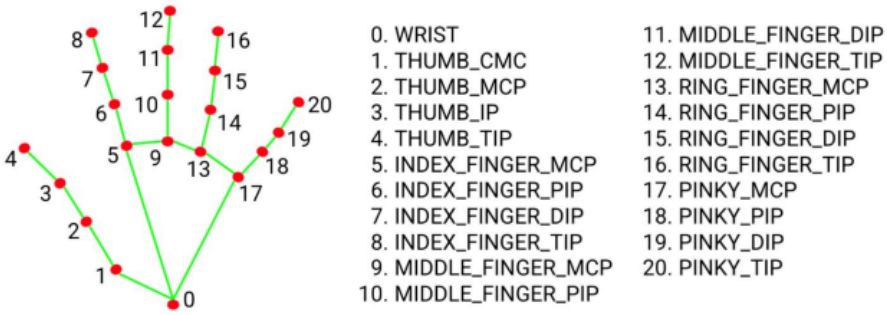


Fig. 2 Mediapipe Hand Tracking Module Landmarks

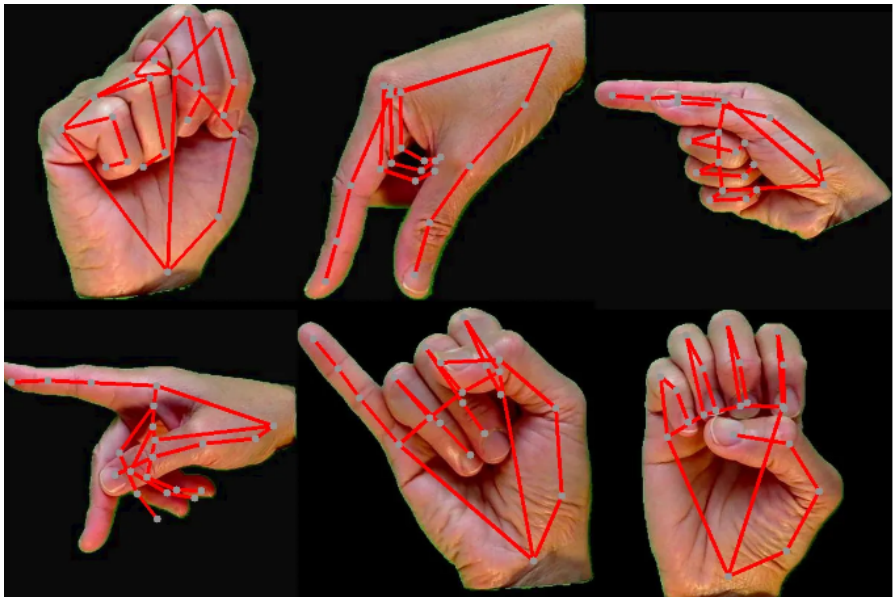


Fig. 3 sign language detection using mediapipe

2.2 SSD Mobilenet V2

SSD MobileNet V2 is a popular object detection model which combines 2 deep learning architectures, MobileNet V2 and SSD (stands for single shot detector). MobileNet V2 is known for its speed, efficiency, and high accuracy for object detection. SSD is an object detection framework that can detect multiple objects in an image with high accuracy and efficiency. SSD MobileNet V2 of image size 320x320. Fig 4 shows its architecture.

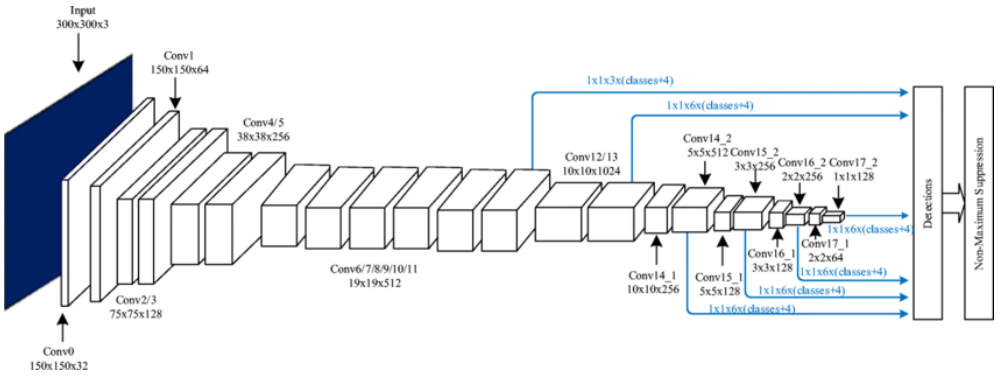


Fig. 4 SSD MobileNet V2 architecture

The SSD MobileNet V2 architecture consists of two main components: a MobileNet V2 backbone network and a Single Shot Detector (SSD) object detection network. The MobileNet V2 backbone network is a lightweight convolutional neural network that is designed to perform well on mobile devices with limited computational resources. The 53 convolutional layers that make up the backbone network include depth-wise separable convolutions, allowing for effective feature extraction with fewer parameters. The input image is utilized to extract features from the MobileNet V2 backbone network, which has been pre-trained on a sizable dataset of photos.

The SSD object detection network is used to detect the objects in the image by predicting the bounding boxes and class labels for each object. In this case, Label maps were created to make the model map between the alphabets, for example: For the alphabet "A", ID was initialized 0, so when the model is trained with the images of A and the test is carried out, model maps to the said ID in this instance "0" hence that's how the model knows how to classify images for all sets of alphabets. The SSD network has a number of detection layers that can identify objects with various scales and aspect ratios

. The MobileNet V2 backbone network's feature extraction layers are connected to the detection layers, enabling them to extract features from the input image. The SSD network consists of multiple layers, some of them being the Convolutional Layer: The convolutional layers are used to extract features from the input images. Activation Layer: The activation layers are used to introduce non-linearity into the network. Prediction Layer: The prediction layers are used to predict the class labels and bounding box coordinates for each image in that object.

3 Methodology

3.1 YOLOv5

YOLOv5 [5] is a deep learning-based architecture used in object detection tasks. Compared to various state-of-the-art models, YOLOv5 offers a compelling balance between speed and accuracy while requiring less processing power. Notably, YOLOv5 was the first open-source implementation released by Glenn Jocher, independent

of a supporting research paper [1]. The project is maintained on GitHub (https://github.com/ultralytics/yolov5) and is described as "ongoing development."

Developed using Python, YOLOv5 prioritizes ease of setup and integration, particularly for deployment on Internet of Things (IoT) devices. Additionally, the PyTorch framework provides a more active development community compared to the Darknet framework used in earlier YOLO versions. YOLOv5 achieves its speed advantage through the use of CSPDarknet53 as its backbone for feature extraction and the Path Aggregation Network (PANet) for enhancing information flow within the model (see Fig 5).

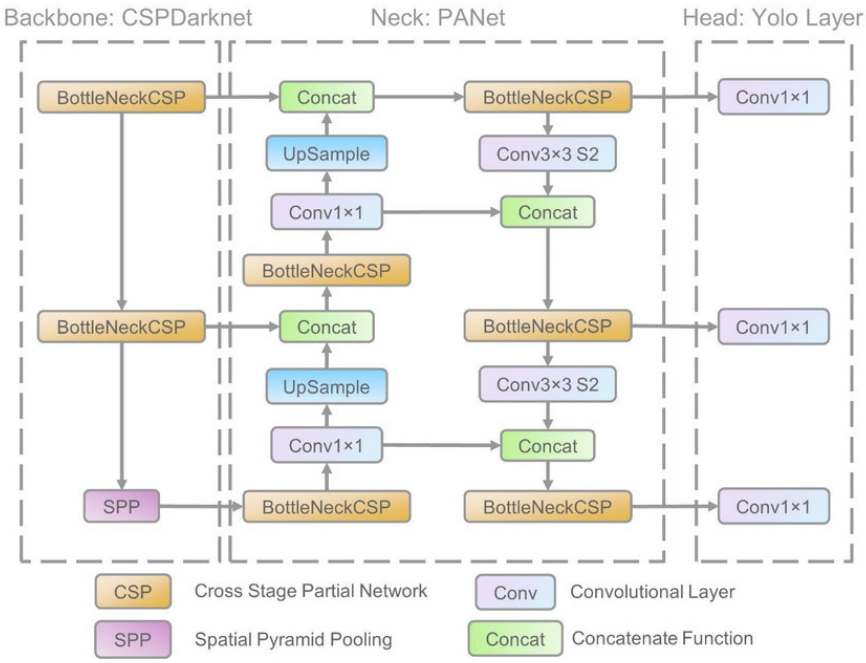


Fig. 5 YOLOv5 architecture

The following reasons support the use of YOLOv5:

1) The YOLOv5 contains SOTA components such as an activation function, hyperparameter, data augmentation approach, and an easy-to-follow user manual.

2) The model's simple architecture enables computational convenience during schooling, even with minimal resources.

3) Yolov5's small size and light weight make it ideal for embedded and cellular applications.

3.2 YOLOv8

YOLOv8, released in January 2023 by Glenn Jocher [6], is on of the latest iterations in the YOLO series of object detection models. This cutting-edge computer vision model is designed for real-time applications and boasts new features like mosaic data augmentation and anchor-free detection. A user-friendly command-line interface (CLI) simplifies model setup, while a Python package streamlines the development process compared to previous versions. (Source: GitHub ultralytics/ultralytics).

YOLOv8 operates by first analyzing an image. It then simultaneously predicts bounding boxes (areas around objects) and class probabilities (the likelihood of an object belonging to a specific class). To achieve this, the image is divided into a grid of equal-sized cells. YOLOv8 predicts confidence scores for each cell's bounding boxes, indicating the possibility of an object existing within that box.

Due to its speed and accuracy, YOLOv8 is a powerful tool for real-time object detection tasks. One potential application is hand gesture recognition in video outlines.

The accuracy of any object detection system, including one using YOLOv8, hinges on several factors. These factors include the quality and quantity of training data used, the model architecture itself, and hyperparameter tuning (configuration settings for the model). Below Fig 6 shows YOLOv8 architecture.

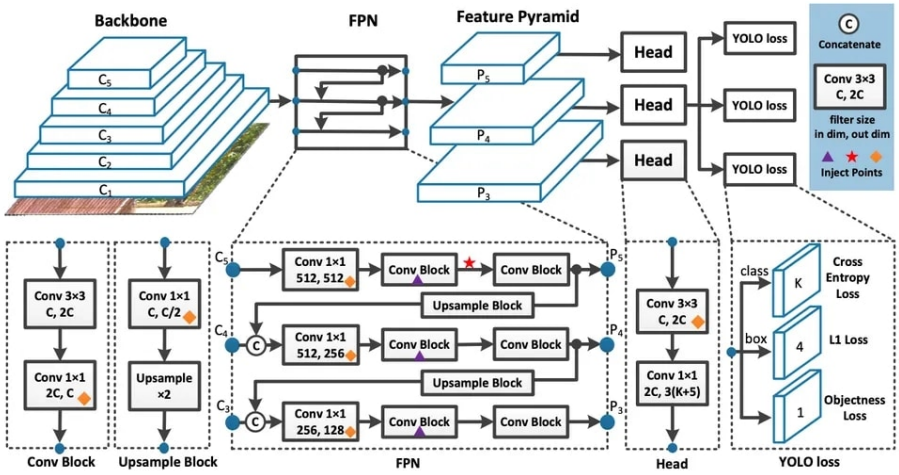


Fig. 6 YOLOv8 architecture

3.3 Training process

In this research, we trained both the medium versions of YOLO v5 and YOLO v8 for sign language recognition.

Key Training Parameters.

- **Image size:** We resized all images in the training dataset to a uniform size of 640x640 pixels to ensure consistency for the model's input.
- **batch size:** we used 16 during training.
- **Number of epochs:** For YOLO v5, we used 150 epochs, while for YOLO v8, we used for 100 epochs.

Our YOLO v5 and YOLO v8 models achieved superior performance compared to approaches utilizing MediaPipe and the SSD-MobileNet-V2 model. Yolo can overcome the challenges of other object detection models as it offers many features:

1-**Single Shot Detection**: Yolo is a single-stage object detection model that predicts bounding boxes and class probabilities for objects by processing the full image at once. Because of its efficiency, it can be used for real-time tasks like hand detection.

2-**Robustness to Size Variations**: YOLO is made to recognize items in an image at various scales. It effectively handles item size fluctuations by using a grid-based method to anticipate bounding boxes and class probabilities throughout the entire image.

3-**End-to-End Training**: Yolo is trained end-to-end, meaning it acquires the ability to simultaneously optimize the tasks of localization and classification. This may help improve robustness and generalization when identifying hands of different sizes and orientations.

4 Dataset

The dataset used as input for the sign language recognition system is a dataset contains American Sign Language characters only sign hands[7] and other data contains characters and words American-sign-language (ASL)[8].

4.1 Sign hands:

Contains 26 classes from A to Z (Fig. 7) and data is split into train, validation, and test. Data specification is given in Table 1.

Table 1 data description

specification	value
Numbers of images	1815
Train data	1271(70%)
Valid data	363(20%)
Test data	181(10%)
Number of classes	26
Number of images per class	70

Fig. 7 Sign hands dataset

4.2 American-sign-language (ASL):

This dataset consists of videos and these videos were split into frames (Fig. 8) every frame was annotated and contains 106 classes (26 classes are characters, 80 classes are words). This data is split into train, validation, and test. Data specification is given in Table 2.

Table 2 data description

specification	value
Numbers of images	23343
Train data	20630(88%)
Valid data	1768(8%)
Test data	945(4%)
Number of classes	106
class	A-B-C-D-E-F-G-H-J-K-L-M-N-O-P-Q-S-T-U-V-X-Y-Z-allergy-bag-barbecue-bill-straw-bitter-additional-bread-coupon-bye-cake–cheese-chicken-coke-cold-cost-credit card-cup-enjoy-dessert-drive-eat-ketchup-egg-fork-French-fries-fresh-hello-hot-bacon-ice cream-ingredients-juicy-lactose-lettuce-mustard-lid-manager-pepper-menu-alcohol-milk-napkin-no-order-cash-pickle-burger-please-ready-refill-repeat-safe-drink-salt–sandwich-sauce-biscuit-small-soda-tissues-sorry-spicy-spoon-sugar-sweet-thank you–tomato-vegetables-total-urgent-receipt-wait-what-water-yoghurt-warm-would–your-pizza

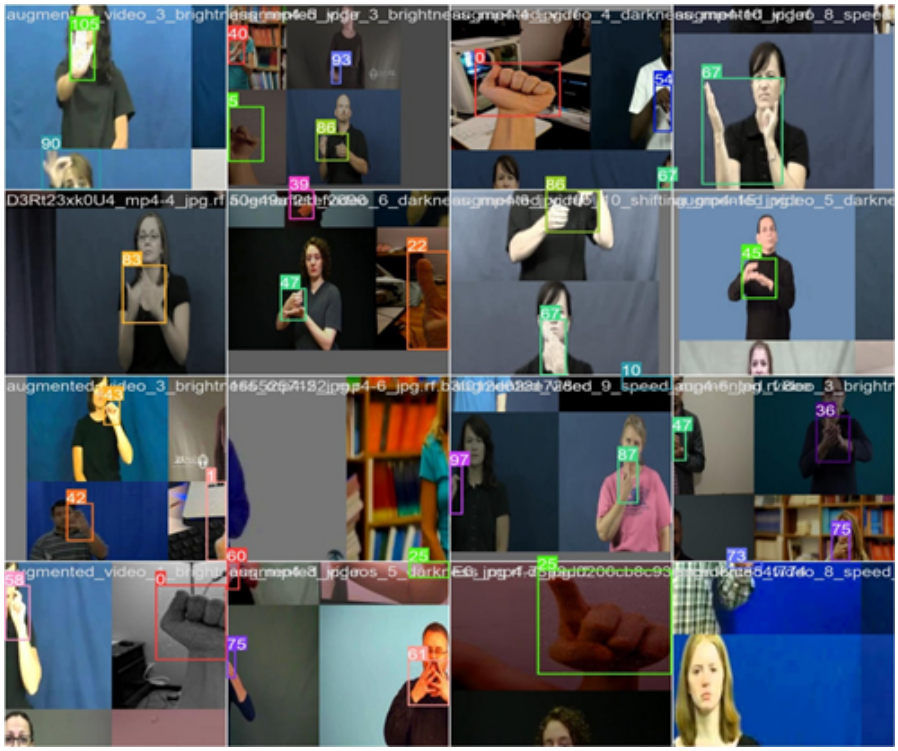


Fig. 8 ASL dataset.

5 Results

We used two different datasets split into training, testing, and validation. Dataset with characters only achieved in Yolo v5 96.7% recall, 96.5% precision, and 98% mean average precision(MAP) whereas achieved in Yolo v8 97% recall, 98% precision, and 99% mean average precision (Table 3).

The second dataset achieved in Yolo v5 95% recall, 94% precision, and 98% mean Average precision whereas achieved in Yolo v8 96% recall, 95% precision, and 98% mean average precision (Table 4).

Fig 13. Shows the confusion matrix of v5 and v8 on the second dataset. Yolo v5 has declined faster than Yolo v8 in both classification loss and bounding box (Fig 14). Yolo v8 is faster and more accurate than Yolo v5 in detecting sign language .

Table 3 results of sign hands dataset

model type	precision	recall	MAP	Box loss	Classification loss
YOLO v5	96.5%	96.7%	98%	0.01	0.009
YOLO v8	98%	97%	99%	0.06	0.09

Table 4 results of American-sign-language (ASL) dataset

model type	precision	recall	MAP	Box loss	Classification loss
YOLO v5	94%	95%	98%	0.01103	0.00135
YOLO v8	95%	96%	98%	0.02	0.066

6 3D

One exciting frontier in sign language translation technology is the use of 3D avatars. While traditional systems may rely on text or 2D animations, 3D avatars offer a more natural and expressive way to convey signs. This section will explore the potential of 3D sign language translation as showed in Fig.9

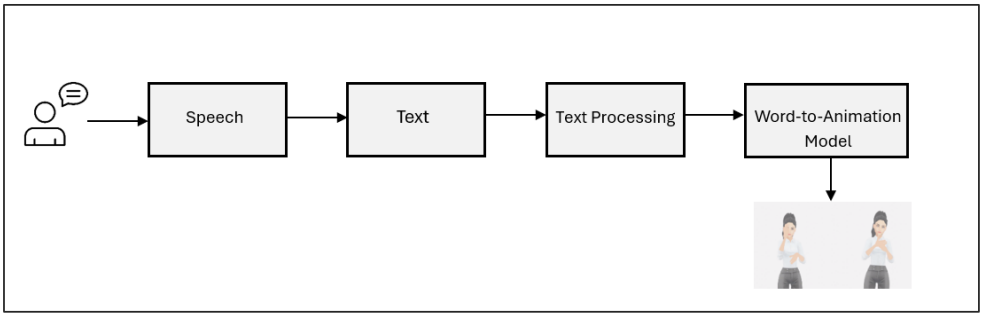


Fig. 9 Generating 3D Avatar Animations from Speech Input.

6.1 Avatar Creation

Blender is an open-source tool for creating 3D animations[9], It is used to execute Python scripts for different functionalities. [10].

- **Modeling**: This involves creating the basic structure of the avatar using modes like pose mode, object mode, and edit mode. Starting with the fundamental structure, such as modeling the fingers before the palm in a human body, the avatar's form is gradually built up (Fig. 10). This process includes shaping meshes, curves, and other elements to form the desired appearance.

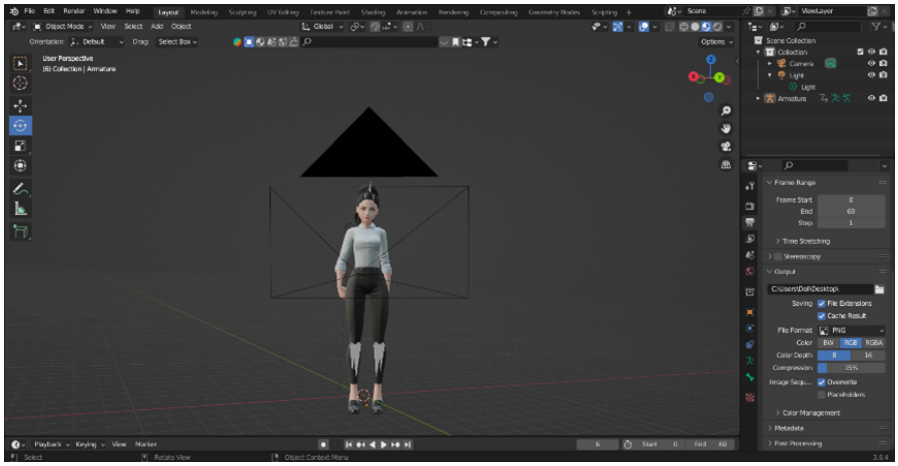


Fig. 10 Avatar Modelling in Blender

- **Rigging**: Once the model is created, bones are added and attached to the structure (Fig. 11). This creates a skeletal framework, which is known as rigging. In rigging, a wireframe surrounds the body to define its boundaries. For example, in a 3D avatar, finger bones would be connected to the mesh representing the hand. Rigging also involves applying constraints to the bones, such as inverse kinematics constraints and bone constraints. Additionally, specific bones may have kinematic constraints applied to them. Euler transforms are commonly used for bone rotation in this stage.

Fig. 11 Avatar Rigging in Blender

- **Animation**: encompasses the dynamic alteration of an object's position or form over a duration. This dynamic transformation can be realized through various techniques, including relocating the object as a unified entity by altering its position, reshaping it by manipulating its individual vertices or control points, and implementing inherited animations where an object's movement is influenced by another object's motion. Keyframes serve as pivotal markers for storing property values at specific time instances. In this system, keyframes are established at intervals of 25 frames within an animation timeline comprising a total of 60 frames. The pace of the animation output can be adjusted to suit desired speed parameters.
- **Rendering**: involves the translation of a 3D scene into a sequence of 2D images, culminating in the creation of a video. In Blender, a diverse array of video formats such as MPEG-1, AVI, MPEG-2, and H.264 (MP4) are supported. Following the animation process, data is organized and stored in individual frames. Rendering encompasses the execution of instructions to assemble these frames into a coherent visual narrative. The resulting animation is subsequently converted into the designated video format. In the context of this system, the rendered video output is formatted as FFmpeg Video and the Video Codec is H.264 with medium quality and good encoding speed.

6.2 Animation script and Time

Blender, equipped with a Python script editor, is utilized in this context to generate animation. There are start and stop keyframes given to every word in the final text translator block's output file. Initially, the avatar is positioned in the default T-pose(Fig. 12).

Fig. 12 T-pose

The beginning and ending positions corresponding to the sign are fetched from memory in respect to the word being processed in the loop during the start and end keyframes. The bones are then given the X, Y, and Z coordinates.

Rendering and creating videos are the two most time-consuming parts of the sign language translator. The command line displays the amount of time needed, which we measured. There is a difference in the amount of time required for text translation between translating longer and shorter sentences:

- Rendering a single word like "sad" takes 70 seconds.
- Generating a 70-frame video, such as" I love you", requires 120 seconds.

The rendering time in the sign language translator is primarily influenced by the following factors:

1. **Complexity of Animation**: The level of detail in the animation directly affects rendering time. More complex animations with intricate movements and effects typically require longer to render.

2. **Number of Frames**: The total number of frames in the animation is a crucial determinant of rendering time. Higher frame counts result in longer rendering durations.

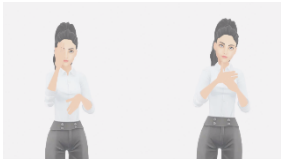
3. **Hardware Resources**: The processing power of the hardware, including CPU and GPU capabilities, as well as available RAM, significantly impacts rendering speed. More powerful hardware generally leads to faster rendering times.

4. **Render Settings**: Parameters such as resolution, quality settings, and rendering techniques influence rendering time. Higher resolutions and quality settings typically require more processing time.

5. **Software Optimization**: The efficiency and optimization of the rendering software also play a vital role. Well-optimized rendering engines can substantially reduce rendering times compared to less optimized ones.

By optimizing these key factors, developers can effectively manage and minimize rendering times in the sign language translator. To illustrate the 3D Avatar's capabilities, Table 5 presents a sample of English words and their corresponding sign language translations.

Table 5. English word-to-sign animation

English Word	Sign Movement	English Word	Sign Movement
Stop		Finish	
Play		Angry	

7 Figures

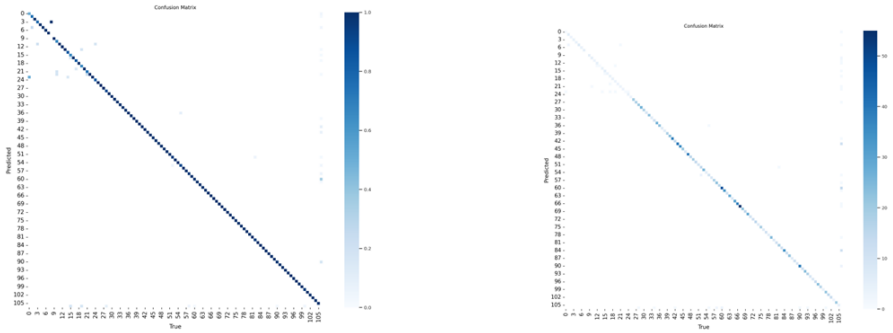


Fig 13. Confusion matrix for YOLOv5(left) and YOLOv8(right)

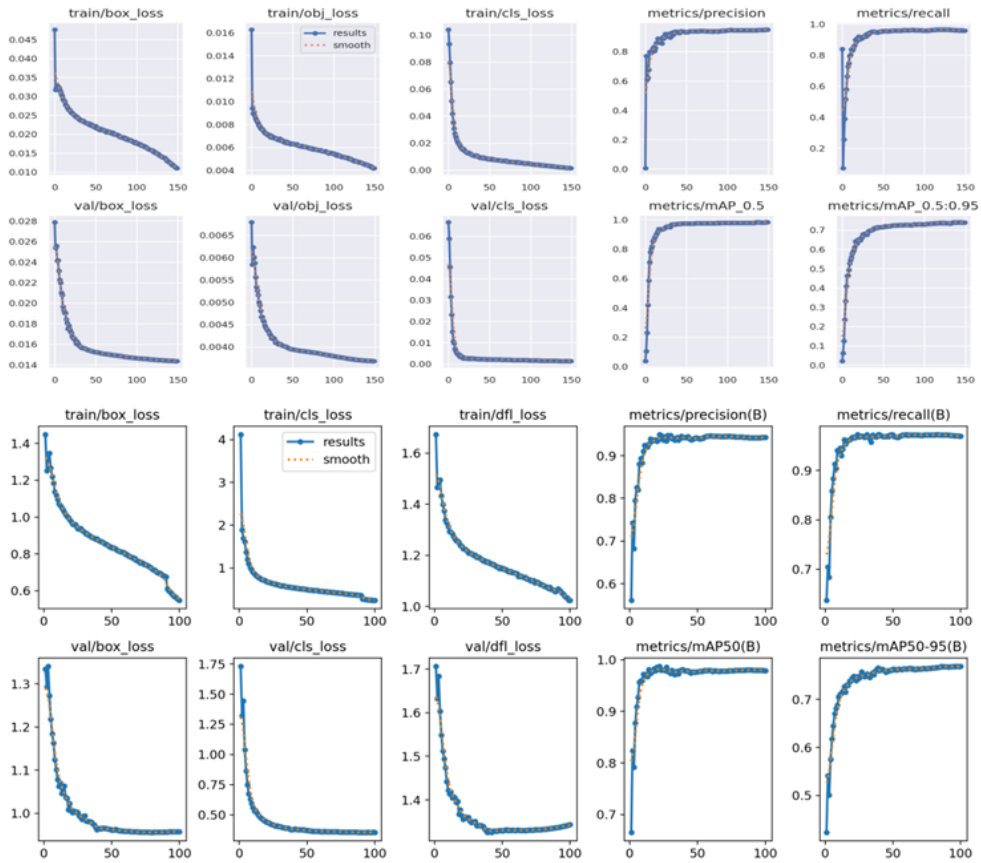


Fig 14. Results and loss reduction for YOLOv5(above) and YOLOv8(below)

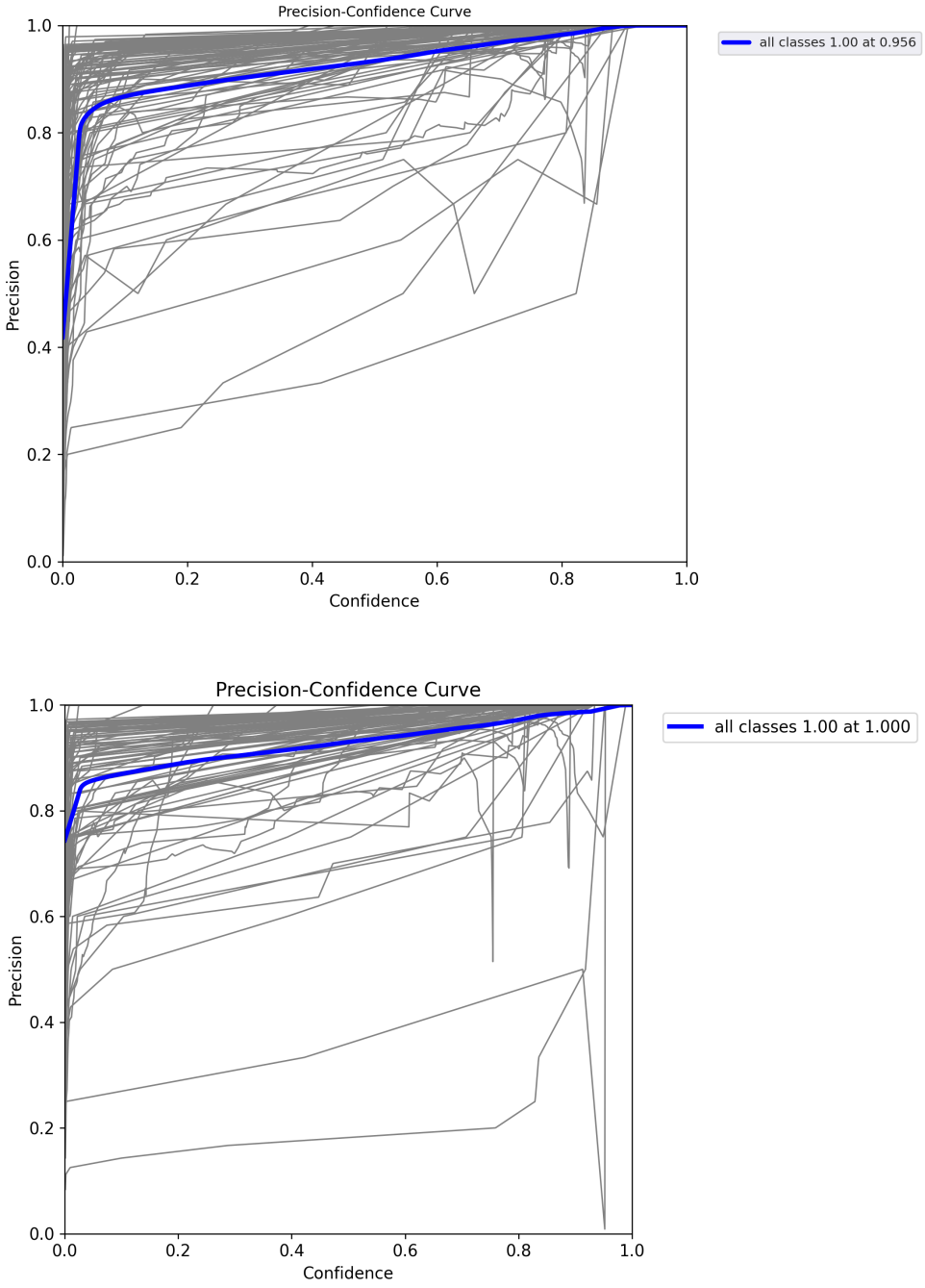


Fig 15. Precision curve for YOLOv5(above) and YOLOv8(below)

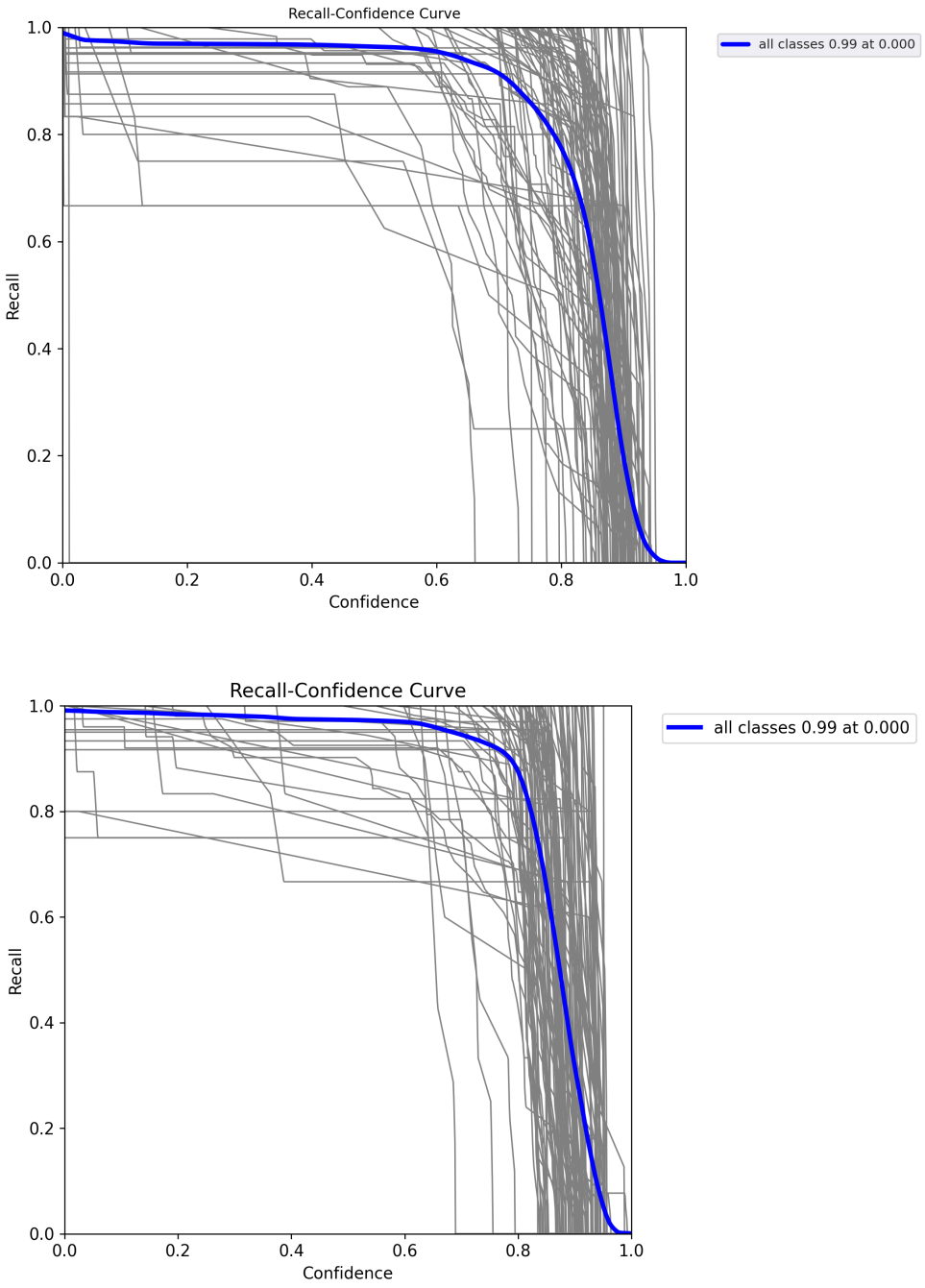


Fig 16. Recall curve for YOLOv5(above) and YOLOv8(below)

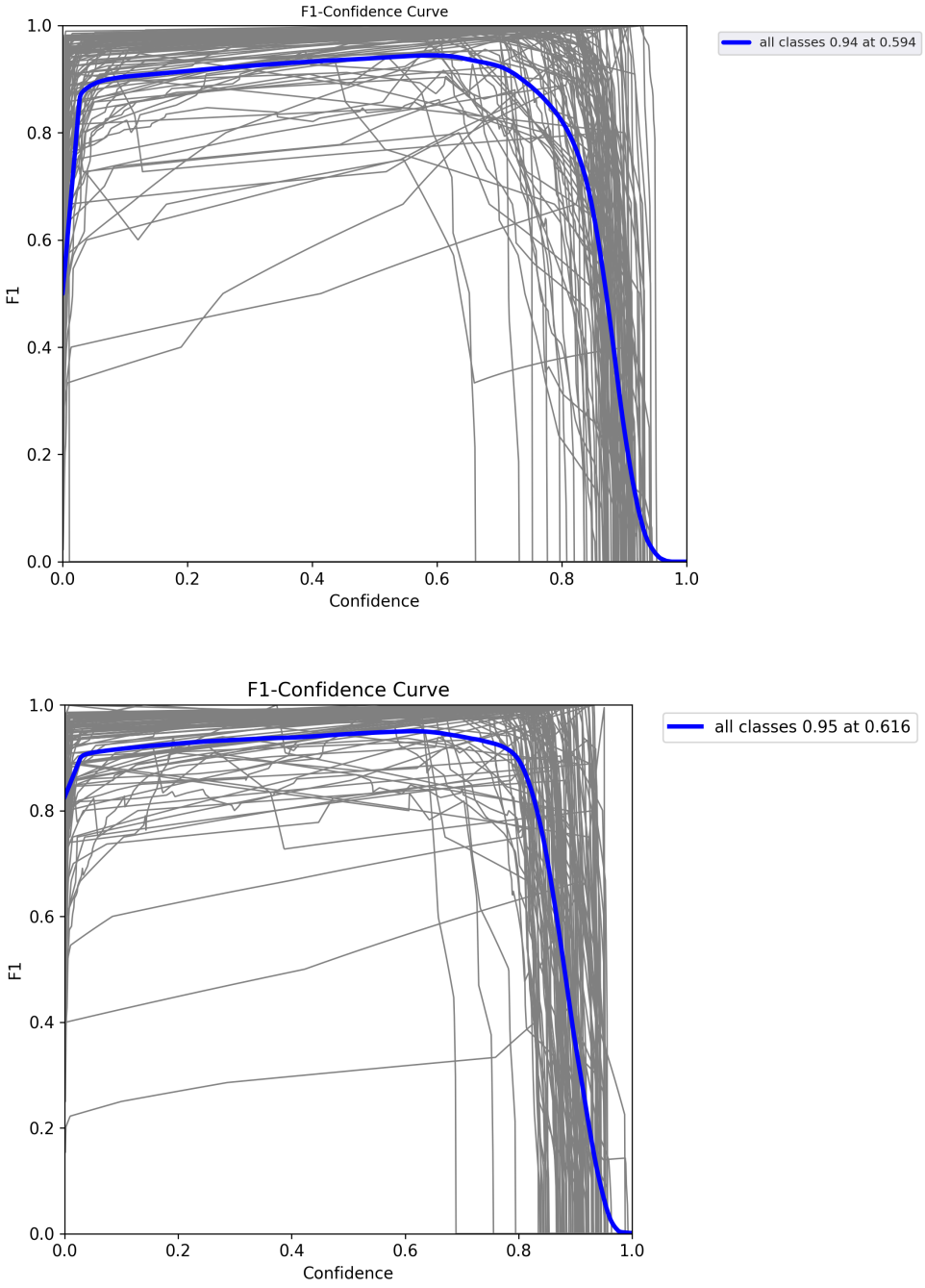


Fig 17. F1-confidence for YOLOv5(above) and YOLOv8(below)

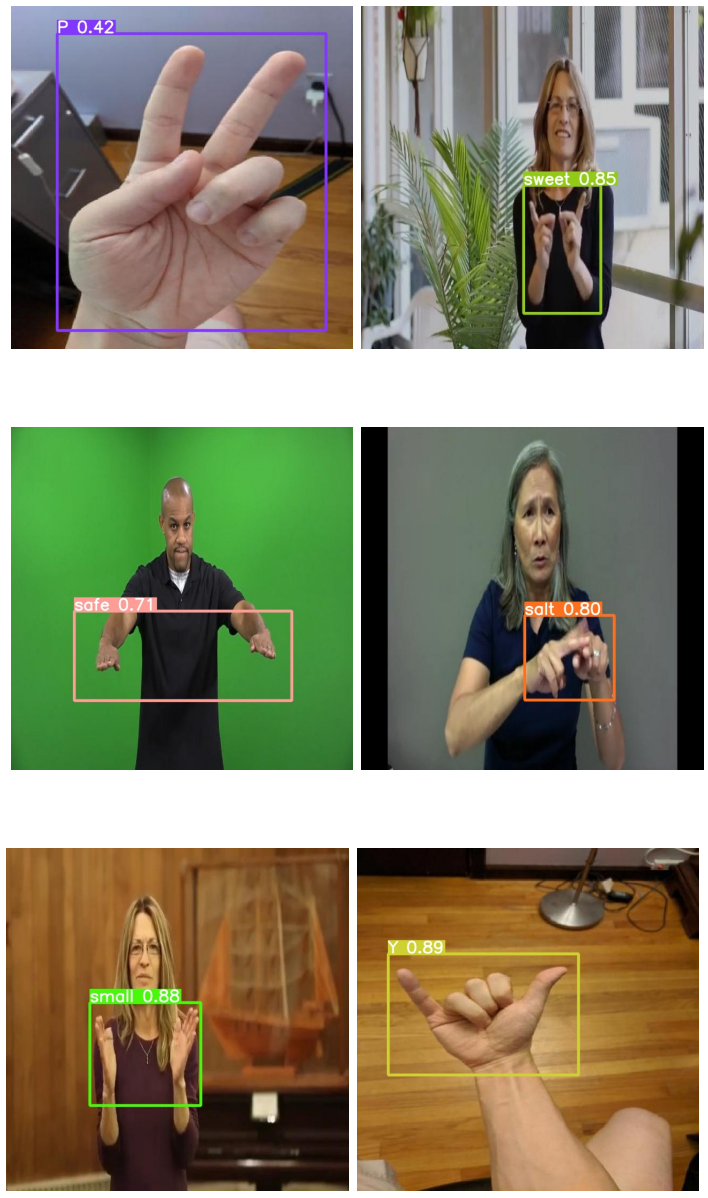


Fig 18. Prediction (YOLO V5)

Fig 19. Prediction (YOLO V8)

8 Conclusion

This research aims to address the communication barrier between deaf and hearing individuals by developing a vision-based American Sign Language (ASL) recognition system using YOLO architecture. The research successfully explored the development of a mobile application for two-way communication between deaf/mute and hearing individuals. The application leverages YOLO v8 and YOLO v5 deep learning architectures to achieve sign language recognition and translation.

Key Findings.

- The study demonstrates the effectiveness of YOLO architectures for sign language recognition. YOLO v8 achieved superior performance compared to YOLO v5 on both datasets, reaching a maximum precision of 98%, recall of 97%, and mean Average Precision (MAP) of 99% for character recognition.
- The application exhibits promising results for recognizing both individual characters and words in American Sign Language (ASL).

Acknowledgements. We would like to express our sincere gratitude to Eng. Abdel Maula Youssef for his invaluable support throughout this research project. His contributions significantly enhanced the quality of this research.

Future Work

- **Expanding the ASL sign vocabulary** within the application for broader communication capabilities.
- **Embedding Hardware for Enhanced Accessibility**. This might include wearable devices like gloves or sensors that can capture sign language data with greater precision and potentially improve recognition accuracy in various lighting conditions or hand positions.
- **Multilingual Support for Global Accessibility**. To broaden its reach and foster communication across international borders, future development will involve incorporating additional sign languages.
- **Integration of Augmented Reality (AR) Technology**. Imagine a user holding their phone up and seeing corresponding ASL signs displayed as virtual annotations alongside a spoken or written sentence. This immersive approach could revolutionize sign language education and communication for both deaf/mute and hearing users

References

[1] Kyle, Jim G., James Kyle, and Bencie Woll. Sign language: The study of deaf people and their language. Cambridge university press, 1988.

[2] Padden, Carol, and Claire Ramsey. "American Sign Language and reading ability in deaf children." Language acquisition by eye 1 (2000): 65-89.

[3] Saiful Bahri, Iffah Zulaikha, et al. "Interpretation of Bahasa Isyarat Malaysia (BIM) Using SSD-MobileNet-V2 FPNLite and COCO mAP." Information 14.6 (2023): 319.

[4] Harris, Moh, and Ali Suryaperdana Agoes. "Applying hand gesture recognition for user guide application using MediaPipe." In 2nd International Seminar of Science and Applied Technology (ISSAT 2021), pp. 101-108. Atlantis Press, 2021.

[5] Wu, Wentong, et al. "Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image." PloS one 16.10 (2021): e0259283.

[6] Terven, Juan, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. "A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas." Machine Learning and Knowledge Extraction 5.4 (2023): 1680-1716.

[7] https://universe.roboflow.com/atuzim/american_sign_language-sgkct

[8] https://universe.roboflow.com/majorproject-25tao/american-sign-language-v36cz

[9] Blender 4.1 Reference Manual. https://docs.blender.org/manual/en/latest/

[10] Blender Foundation, "Game Engine - Blender Manual," Blender Foundation, [Online]. Available: https://docs.blender.org/manual/en/dev/game_engine/index.html

[11] Flavell, Lance. Beginning blender: open source 3d modeling, animation, and game design. Apress, 2011.

[12] Das Chakladar, Debashis, Pradeep Kumar, Shubham Mandal, Partha Pratim Roy, Masakazu Iwamura, and Byung-Gyu Kim. 2021. "3D Avatar Approach for Continuous Sign Movement Using Speech/Text" Applied Sciences 11, no. 8: 3439. https://doi.org/10.3390/app11083439

[13] Petkar, Tanmay, Tanay Patil, Ashwini Wadhankar, Vaishnavi Chandore, Vaishnavi Umate, and Dhanshri Hingnekar. "Real Time Sign Language Recognition System for Hearing and Speech Impaired People." Int J Res Appl Sci Eng Technol 10, no. 4 (2022): 2261-2267.

[14] Adamo-Villani, Nicoletta. "3d rendering of american sign language finger-spelling: a comparative study of two animation techniques." International journal of human and social sciences 3, no. 4 (2008): 24.

[15] Alaftekin, Melek, Ishak Pacal, and Kenan Cicek. "Real-time sign language recognition based on YOLO algorithm." Neural Computing and Applications (2024): 1-16.

[16] Patel, M. (2023). American sign language detection (Doctoral dissertation, California State University, Northridge).

[17] Halder, Arpita, and Akshit Tayade. "Real-time vernacular sign language recognition using mediapipe and machine learning." Journal homepage: www. ijrpr. com ISSN 2582 (2021): 7421.

[18] 1.Elakkiya R (2021) Machine learning based sign language recognition: a review and its research frontier. J Ambient Intell Humaniz Comput 12:7205–7224

[19] Vidhyasagar, B. S., An Sakthi Lakshmanan, M. K. Abishek, and Sivakumar Kalimuthu. "Video Captioning Based on Sign Language Using YOLOV8 Model." In IFIP International Internet of Things Conference, pp. 306-315. Cham: Springer Nature Switzerland, 2023.

[20] Lui, Michael Stephen, and Fitri Utaminingrum. "A Comparative Study of YOLOv5 models on American Sign Language Dataset." Proceedings of the 7th International Conference on Sustainable Information Engineering and Technology. 2022.

[21] Bora, Jyotishman, et al. "Real-time assamese sign language recognition using mediapipe and deep learning." Procedia Computer Science 218 (2023): 1384-1393.

[22] Shams, Mahmoud Y., et al. "Food Item Recognition and Calories Estimation Using YOLOv5." International Conference on Computer Communication Technologies. Singapore: Springer Nature Singapore, 2023.

[23] Shams, Mahmoud Y., Omar M. Elzeki, and Hanaa Salem Marie. "Towards 3D virtual dressing room based user-friendly metaverse strategy." The Future of Metaverse in the Virtual Era and Physical World. Cham: Springer International Publishing, 2023. 27-42.